

CBAP® **Certification** **and BABOK®** **Study Guide**

Hans Jonasson



CRC Press
Taylor & Francis Group

AN AUERBACH BOOK

CBAP®

Certification

and BABOK®

Study Guide



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

CBAP®

Certification

and BABOK®

Study Guide

Hans Jonasson



CRC Press

Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
AN AUERBACH BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2017 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed on acid-free paper
Version Date: 20160726

International Standard Book Number-13: 978-1-4987-6725-5 (Hardback)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the CRC Press Web site at
<http://www.crcpress.com>

Contents

ABOUT THE AUTHOR	xi
CHAPTER 1 INTRODUCTION	1
1.1 Objectives	1
1.2 Overview	2
1.3 Early Days	3
1.4 Project Management Institute®	5
1.5 International Institute of Business Analysis™	6
1.6 Role of the Business Analyst	7
1.7 Where Is It All Going?	8
1.8 Book Project	9
1.9 Summary	9
Activity	10
CHAPTER 2 LAYING THE FOUNDATION	11
2.1 Objectives	11
2.2 Overview	11
2.3 Life Cycle Definitions	12
2.4 What Is a Body of Knowledge?	14
2.5 Overview of PMI Applicable Standards	15
2.6 Overview of IIBA Framework and Standards	17
2.7 SEI-CMMI and Applicability	19
2.7.1 Requirements Management	20
2.7.2 Requirements Development	21
2.8 Which Standard to Use?	21
2.9 Comments on Tool Standards	22
Activity	23
CHAPTER 3 BABOK OVERVIEW	25
3.1 Objectives	25
3.2 Overview	25
3.3 Key Concepts	25

3.4	Perspectives	26
3.4.1	Agile	27
3.4.2	Business Intelligence	27
3.4.3	Information Technology	27
3.4.4	Business Architecture	27
3.4.5	Business Process Management	28
3.5	Business Analyst's Skills	28
3.5.1	Analytical Thinking and Problem Solving	28
3.5.2	Behavioral Characteristics	29
3.5.3	Business Knowledge	29
3.5.4	Communication Skills	29
3.5.5	Interaction Skills	30
3.5.6	Tools and Technology	31
3.6	Summary	31
CHAPTER 4 BUSINESS ANALYSIS PLANNING AND MONITORING		33
4.1	Objectives	33
4.2	Overview	33
4.3	IIBA View	34
4.3.1	Plan the Business Analysis Approach	34
4.3.1.1	Overview of Approaches	36
4.3.1.2	Selecting and Customizing a Process	37
4.3.1.3	Waterfall	38
4.3.1.4	Iterative	41
4.3.1.5	Agile	43
4.3.2	Plan Stakeholder Engagement	49
4.3.2.1	User Profiling	52
4.3.2.2	Experienced or Novice Users	53
4.3.2.3	Full-Time or Part-Time Users	54
4.3.2.4	Organizational Entities	54
4.3.2.5	Internal or External Users	54
4.3.2.6	Language and Cultural Differences	54
4.3.2.7	Geographical Differences	55
4.3.2.8	Stakeholder Communication	55
4.3.3	Plan Business Analysis Governance	56
4.3.4	Plan Business Analysis Information Management	57
4.3.5	Plan Business Analysis Performance Improvements	57
4.4	Business Analysis Plan	58
4.4.1	Project Overview and Background	59
4.4.2	Scope and Deliverables	59
4.4.3	Project Activities	60
4.4.4	Roles and Responsibilities	61
4.4.5	Resource Plan	62
4.4.6	Requirements Risk Plan	62
4.4.7	Manage Changes to Requirements	63
4.5	General Guidelines for the Requirements Plan	64
4.5.1	Risk	64
4.5.1.1	Step 1: Develop the Risk Management Approach	64
4.5.1.2	Step 2: Identify Risks	64
4.5.1.3	Step 3: Assess Risks	66
4.5.1.4	Step 4: Respond to Risks	67
4.5.1.5	Step 5: Monitor and Control Risks	67

4.5.2	Estimating	68
4.5.3	Laying Out Tasks	68
4.5.4	Costing	69
4.5.5	Tracking and Reporting	70
4.5.6	Kickoff Meeting	71
4.5.7	Summary	71
	Activity	72
CHAPTER 5	ELICITATION AND COLLABORATION	73
5.1	Objectives	73
5.2	Overview	74
5.3	IIBA View	74
5.4	How to Select the Right Technique for Gathering Requirements	76
5.4.1	Different Ways for Different Customers	77
5.4.2	Different Ways for Different Categories	77
5.4.3	Impact of Globalization	78
5.5	Customer Interviews	79
5.6	Observation or Job Shadowing	87
5.7	Studying Existing Systems	89
5.8	Studying Interfaces	90
5.9	Surveys	92
5.10	Discovery/JAD/Facilitated Sessions	94
5.10.1	History	95
5.10.2	Characteristics of a JAD Project	95
5.10.3	What Is Created in a JAD Session?	95
5.10.4	JAD Participants	96
5.10.4.1	Facilitator	96
5.10.4.2	Business Analyst	98
5.10.4.3	Scribe	98
5.10.4.4	User (Customer)	99
5.10.4.5	Subject-Matter Experts	99
5.10.4.6	Developer	100
5.10.4.7	Sponsor	100
5.10.4.8	Observers	101
5.10.5	JAD Process	101
5.10.5.1	Establish Goals and Objectives	101
5.10.5.2	Prepare for the Session	101
5.10.5.3	Conduct the Session	103
5.10.5.4	Follow-Up	107
5.10.6	Facilitation	107
5.10.7	Summary	109
5.11	Focus Groups	110
5.12	Market Research	111
5.13	Evaluate <i>Best Practices</i>	112
5.14	Prototyping	113
5.15	Storyboarding	115
5.16	Idea-Generating Techniques (Brainstorming)	117
5.16.1	Basic Brainstorming	117
5.16.2	Anonymous Brainstorming	119
5.16.3	Affinity Diagramming	119
5.16.4	Brainstorming with One Person	120
5.16.5	Brainstorming for a Virtual Team	121
5.16.6	Summary	121

5.17	Tools for Virtual Environments	122
5.18	Requirements Prioritization Techniques	122
5.18.1	Dollar Approach	123
5.18.2	Forced-Pair Ranking	124
5.18.3	Density Dotting	125
5.18.4	Analytical Hierarchy Process	126
5.18.5	Summary	126
5.19	Prepare Requirements Package	126
5.20	Communicate Requirements	131
5.21	Summary	132
	Activity	133
CHAPTER 6 REQUIREMENTS LIFE CYCLE MANAGEMENT		135
6.1	Objectives	135
6.2	Overview	135
6.3	IIBA Tasks	136
6.3.1	Trace Requirements	136
6.3.2	Maintain Requirements	137
6.3.3	Prioritize Requirements	137
6.3.4	Assess Requirements Changes	138
6.3.5	Approve Requirements	138
6.4	Summary	139
	Activity	139
CHAPTER 7 STRATEGY ANALYSIS		141
7.1	Objectives	141
7.2	Overview	141
7.3	IIBA Tasks	143
7.3.1	Define the Business Need	145
7.3.2	Assess Capability Gaps	150
7.3.3	Determine the Approach to Finding a Solution	152
7.3.4	Define Solution Scope	152
7.3.4.1	Business Goals and Objectives	153
7.3.4.2	Assumptions	153
7.3.4.3	Constraints	154
7.3.4.4	Scope Statement	154
7.3.4.5	Impacted Organizations	154
7.3.5	Define the Business Case	156
7.4	Conducting the Initial Risk Assessment	157
7.5	Preparing the Decision Package	158
7.6	Understanding the Business	159
7.7	Business Models	160
7.7.1	Organization Charts	160
7.7.2	Infrastructure Models	161
7.7.3	Business Location Models	162
7.7.4	Business Events	162
7.7.5	Business Entity Models	164
7.7.6	Business Process Models	165
7.8	Summary	166
	Activity	167

CONTENTS

IX

CHAPTER 8 REQUIREMENTS ANALYSIS AND DESIGN DEFINITION	169
8.1 Objectives	169
8.2 Overview	170
8.3 IIBA Tasks	172
8.3.1 Specify and Model Requirements	173
8.3.1.1 Process Models	173
8.3.1.2 Data Models	177
8.3.1.3 Data Flow Diagrams	181
8.3.1.4 UML Family	184
8.3.1.5 Matrix Documentation	197
8.3.1.6 How to Decide Which Model to Select	198
8.3.1.7 Text Documentation	198
8.3.2 Verify the Requirements	201
8.3.3 Validate the Requirements	202
8.3.4 Define Requirements Architecture	202
8.3.4.1 How Much Detail Do You Need?	205
8.3.4.2 Stakeholder-Based Classification	207
8.3.4.3 Sequence-Oriented Classification	208
8.3.4.4 Purpose-Based Classification	213
8.3.4.5 Combining Structures	214
8.3.5 Define Design Options	214
8.3.6 Analyze Potential Value and Recommend Solution	218
8.4 Summary	219
Activity	220
CHAPTER 9 SOLUTION EVALUATION	221
9.1 Objectives	221
9.2 Overview	221
9.3 IIBA Tasks	222
9.4 Matching the Solution to the Needs of the Customer	222
9.5 Support Testing and Quality Assurance	223
9.5.1 Validation versus Verification	224
9.5.2 Planning for the Test	224
9.5.3 Types of Tests	227
9.5.4 Evaluating Customer Satisfaction	229
9.6 Implementing and Supporting the Solution	230
9.7 Summary	231
Activity	231
CHAPTER 10 SWEDE-MART CASE STUDY	233
10.1 Introduction	233
10.2 Strategy	233
10.3 Industry Background	234
10.4 Project Background	234
10.5 Distribution Center/Inventory Operations	234
10.6 Product Lines	235
10.7 Purchasing	235
10.8 Receiving	236
10.9 Accounts Payable	236
10.10 Order Processing and Shipping	236
10.11 Reporting	237
10.12 Summary	237
10.13 Interview with Store Buyer	238

CHAPTER 11 CBAP TEST QUESTIONS	239
CHAPTER 12 ACTIVITY SOLUTIONS FOR SWEDE-MART CASE STUDY	291
Chapter 1	291
Actions	291
Meeting	291
Chapter 2	292
Chapter 4	292
Chapter 5	295
Chapter 6	296
Chapter 7	296
Chapter 8	297
Chapter 9	299
CHAPTER 13 PREPARING FOR THE TEST	301
Taking the Test	302
CHAPTER 14 ANSWERS TO TEST QUESTIONS	303
APPENDIX A: ACRONYMS	333
APPENDIX B: BUSINESS REQUIREMENTS DOCUMENT TEMPLATES	335
APPENDIX C: UNITED NATIONS ORGANIZATIONAL CHART	353
INDEX	355

About the Author



Hans Jonasson, CBAP, PMP, founder of JTC Unlimited, has more than 30 years of experience in the areas of project management, business analysis, and professional development training. He started his career with Volvo Ltd. in Gothenburg, Sweden, in 1980 as a systems analyst/programmer. In 1984, he moved to the United States to work on new development projects for EDS and General Motors. He has managed all aspects of software development projects for the automotive industry, with budgets that have ranged from \$100,000 to \$10 million. Hans is now based in Gothenburg, Sweden.

He has taught introductory and advanced-level courses on project management, requirements gathering, Capability Maturity Model Integrated, and process development to more than 20,000 professionals at companies that include IBM, EDS, Ford Motors, DaimlerChrysler, General Dynamics, Citibank, and Volvo.

Hans is a member of the Swedish Chapter of the Project Management Institute and the International Institute of Business Analysis (IIBA). He is also a founding member of the Southeast Michigan IIBA.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

INTRODUCTION

If you don't know where you're going, chances are you will end up somewhere else.

Yogi Berra

In the last 30 years, I have taught project management and requirements gathering to more than 25,000 people worldwide. In most of those classes, there has been discussion about why projects, especially information technology (IT) projects, fail. Inevitably, the number one reason comes back to unclear or changing requirements. When organizations try to address these problems, they often try for quick fixes such as buying new tools or hiring a consultant. However, it takes more than that. Good requirements do not come from a tool, or from a customer interview. They come from a repeatable set of processes that take the project from the early idea stage through to the creation of an agreed-upon project and product scope between the customer and the developer. This repeatable set of processes, and the tools and techniques that help execute them, are what I want to address in this book.

This chapter sets the stage for the rest of the book by getting you familiar with the format, the writing style, and the purpose of the book. Each chapter has a similar structure and format that are used throughout the book. The content is based on certain choices in regard to what standards to use and what techniques to include. These choices were made by me, and are explained in this chapter through a review of the history of systems development and the evolution of today's standards. One of the primary goals in this chapter is to let you know how the book is organized, what it covers, and what is expected of you afterward. This is very much in line with what is recommended when creating a good requirements document.

1.1 Objectives

- Explain the purpose of the book.
- Introduce how to use the book.
- Review the target audience of the book.
- Introduce the professional domain of business analysis.
- Review the evolution of business analysis during the last few decades.
- Review the standard-setting forces in the industry today.
- Introduce the project that will be used throughout the book as an example.

1.2 Overview

This book is designed to be used for multiple purposes:

1. As a study guide for the Certified Business Analysis Professional™ (CBAP®) exam, which will be expanded on later. The text of the book will greatly help in preparing for the exam. Additionally, this book includes 300 sample questions to help you prepare for the exam. It is based on version 3 of the *Guide to the Business Analysis Body of Knowledge®* (BABOK®).
2. As a reference book. Any person involved with gathering requirements can use this book to discover best practices and learn useful techniques, as well as tools and templates, to improve the requirements-gathering processes within his or her organization.
3. For self-improvement. A business analyst needing more guidance on requirements-gathering techniques, or one who is preparing for any certifications in business analysis, can use this book as a study guide. The book contains exercises and sample solutions to demonstrate how to deal with different situations that may be encountered in the requirements-gathering process.
4. As a course book. Study groups, as well as participants in internal corporate training and consultative training, can use this book as a textbook, as it includes built-in exercises, best practices, tools, and templates. Because it also contains comprehensive solutions to the activities, a warning is warranted. The solutions presented are a solution, not the solution. Because there are no real customers being interviewed, there will be assumptions made, so there is likely to be some difference in the book solution versus your solution.
5. To provide templates. Two different examples of a business requirements document have been included in [Appendix B](#) of this book. The first is a comprehensive template, provided courtesy of ESI International, and the second is a simpler template, suitable for smaller projects. Review them, customize them, and introduce them into your organization.

This book is not intended as a tool book nor does it advocate any special methodology or technique. Its purpose is to give a broad overview of a multitude of business analysis issues, and to provide enough information about tools and techniques to enable the reader to select the best approach for different types of projects. It is a generalist book, not a specialist one. For the key areas where an organization may invest a significant amount of time and effort, more comprehensive training should be considered.

To evaluate the current state of the industry and gather pertinent information on the challenges facing the business analysts of today, I conducted a study with a group of business analysts and project participants who had previously attended my training seminars. While not a large or comprehensive study, it served its purpose well by exploring the role of the business analyst in different organizations, the difficulties in

capturing requirements, and information about the tools used by practitioners to assist in the analysis process. This study is referenced and discussed throughout the book.

One of the questions asked on the survey was, “What are the main difficulties with gathering and documenting requirements?” The top five responses were

1. Lack of time or availability on the part of the customer
2. Lack of customer knowledge
3. Lack of buy-in to scope
4. Lack of skilled business analysts
5. No repeatable process from which to learn

These problems, and others, are explored throughout the book. However, at a high level, it is striking that the first three areas all deal with customer issues. Is that because most people responding to the survey were developers, and it is easier to blame the customer? Or is it because customers truly have challenges in the areas of communication and perhaps in their understanding of what the business really needs? It is likely a bit of both. Either way, it clearly shows the importance of the customer’s role and competency in requirements gathering.

1.3 Early Days

In the early days of computer systems, most applications were geared toward solving a specific problem. The customer was often personally known by the developer and had a relatively clear understanding of what was wanted, and the biggest constraint was typically the capabilities of the computers themselves. Hardware was expensive, with limited capabilities. It is always an amazing journey to review what computers were capable of 30 years ago versus what they are able to perform today. And although their capacity continues to improve, the capability of computers is rarely the limiting factor anymore. Back then, computer systems were viewed as tools, like an adding machine or telephone, with no real vision of how the tool could drive organizational change or move the business in new directions. Integration and interoperability were crude at best. Life for the developer was easy because customers were not very picky and normally were quite happy with any tool that would help them in their day-to-day work. It is interesting to draw a parallel with the early days of television: the programming was sparse, the picture quality low, and many of the shows crude (at best), but in general, people were happy with it because expectations were low and not difficult to meet. Today, with hundreds of channels and digital quality, it seems like most people still have a hard time finding anything they like. Expectations have changed.

Although there were several systems development methodologies around in the mid-1980s, they all tended to have a strong focus on the systems side, often at the expense of understanding the real business need. The creation of the system, or programming of an existing system, was typically the starting point of the project, followed by a multitude of trials and errors. Many projects were canceled and even more were not received

well by the customers. There was little or no attention paid to making sure that the developers understood what the actual business problem was that they were trying to solve. Although this was understandable because there was limited precedence in the way of system solutions, customers were often faced with not knowing how to describe what they needed and quite often took the “I’ll know it when I see it” approach.

Gradually throughout the 1980s and 1990s, this started changing. Systems rapidly became more complex, and business more competitive and global, leading to a shift from struggling with what technology could do to struggling with what the business was trying to accomplish. One of the first projects that I worked on after moving to the United States in 1984 was a common accounts payable system for General Motors (GM). Before that, the projects I had seen at Volvo Ltd. in Sweden were mostly single-user, relatively nonintegrated applications. At GM, there was a stronger emphasis on capturing what the customer needed, trying to get consensus between different organizations to build *common* systems, as well as trying to closely integrate the accounts payable function with receivables and purchasing (Figure 1.1).

To be successful in that environment (and because the system is still running some 30 years later, I think one could argue that the project was successful), it took a different skill set than that of the traditional systems developer. Although technical skills are still important, systems development now has become more and more about communication, facilitation, negotiation, and basic people skills. There were many very good technical people who did not enjoy this new environment, resulting in many issues between the developers, with their minds on the solution, and the users, with their minds on what the business needed.

Although technical skills are still important, systems development now has become more and more about communication, facilitation, negotiation, and basic people skills.

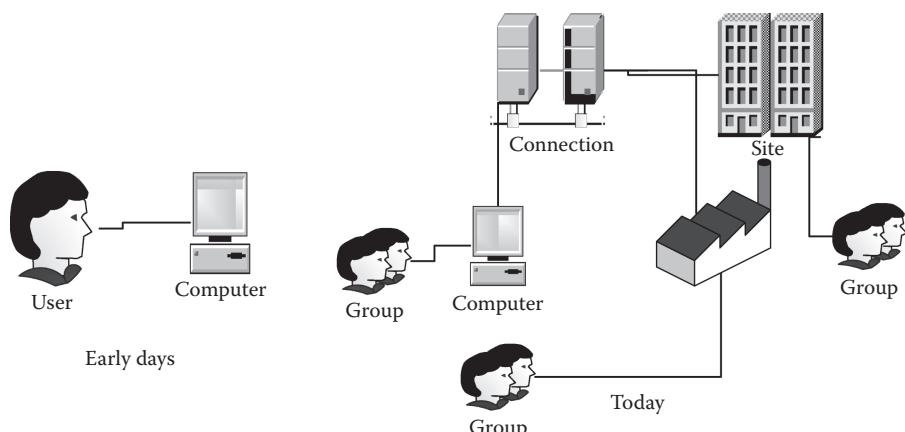


Figure 1.1 Evolution of systems.

In an attempt to try different ways to improve communication between developers and customers, some organizations created account teams made up of analysts responsible for interfacing with the customers, gathering their requirements, and making sure that they were satisfied. Instead of the customer talking to the developer, it became the responsibility of the account team to make sure the development team understood the customer's requirements. Unfortunately, people put in those positions often did not have a strong business background or a good understanding of what the developers needed. On the basis of the lack of tools and training, the business analysts were struggling with this dual communication role—talking business to the customer and systems to the developers. It made projects more complex by requiring coordination between developers, users, and go-betweens. It made the need for good project management practices even more critical than before.

1.4 Project Management Institute®

In the late 1960s, the Project Management Institute (PMI) was started. Although its impact was gradual in the beginning, by the mid-1990s the PMI had become a driving force in the area of establishing standards for project management. Their *Guide to the Project Management Body of Knowledge*, currently in its fourth edition, has been instrumental in establishing a common language and a concentration on the planning aspects of a project.

With the help of the PMI and various development methodologies, the late 1990s saw good development practices and good management of the development process put into place, enabling the development teams to develop something faster. However, that still left a big void. The PMI's domain is how to take an idea and implement it. The product description (in whatever level of detail it is available) is an input into the project management processes. But who creates the product description? How does the developer make sure that the right product is developed? Obviously, it is ideal to develop a solid product and to do it in an organized fashion; however, if it is the wrong product, there is still a big problem. Some people feel that the PMI could have done more to define standards for this, but there is actually a need to separate the role of the project manager and the role of the product developer. They have conflicting roles and conflicting goals. In addition, the product-definition aspect tends to be very dependent on the industry for which the work is being done. Product definition for IT projects is not the same as product definition for construction. Thus, the PMI left the process of requirements development to other organizations. However, recently, the PMI entered into this arena with their PMI Professional in Business Analysis® (PMI-PBA®) certification. While this book is not making any attempt to align with this certification, it would be safe to say that many of the concepts covered in this book are consistent with the PMI view.

1.5 International Institute of Business Analysis™

In 2003, the International Institute of Business Analysis (IIBA) was started in Canada with the goal of improving the working environment for people who are involved in analysis for systems, business, and process improvement. The IIBA released *BABOK* version 3 in 2015. This book is intended to accompany the *BABOK* for people seeking the certification *CBAP*.

The IIBA has had a lot of immediate success and has been embraced by a large number of practitioners, often based on faith alone, which is the way it must be in the beginning. There has been a feeling among business analysts that too much attention has been paid to the project plan and the system design but not enough to defining the product being delivered. The processes and methodologies used have often been rooted in a tool, or connected with a specific vendor. The IIBA is likely to be a key driver in establishing the framework of the professional domain that is being called business analysis.

There has been a feeling among business analysts that too much attention has been paid to the project plan and the system design and not enough to defining the product being delivered.

The role of the business analyst has evolved over time. Initially being a part of the developer's job, the role was known as systems analysis and requirements analysis. It was a job where the duties varied greatly depending on the organization and the tools that were used for development. Now it is evolving into a value-added position, serving as a communication link between the developers and the customers. The language of development can often be cryptic and full of acronyms. So is the language of business. What the organization is looking for in the business analyst is the ability to communicate effectively with both the business organization and the developer. This has become an even larger issue with the offshoring of IT projects. The communication problems traditionally faced by customers and developers, who may be located on different floors and use different terminology, get multiplied when complexities such as time, language, and cultural differences are added. Because globalization is likely to be a trend that is here to stay, there will continue to be an increased need to deal with these issues in the future. The job of the business analyst is secure and appears to be moving in the direction of increased importance and stature. It is actually one of the more difficult jobs to offshore because there is a need to be close to the customer. This by itself may explain, at least partially, the increased interest in business analysis not only from organizations but also from the individual practitioners.

The job of the business analyst is secure and appears to be moving in the direction of increased importance and stature.

1.6 Role of the Business Analyst

In the survey mentioned earlier in this chapter, the respondents were asked about the role of the business analyst in their organization. It was clear from their responses that there is not yet a universally accepted industry standard for this. Examples of responses include

- Create a document.
- Work with end users to define requirements and then with developers to ensure that those requirements can be translated into code.
- Facilitate meetings.
- Approve test plans and test cases.
- Develop process and procedure documentation.
- Create project plans.
- Act as low-level liaison between customers and IT.
- Train users.
- Create designs for offshore positions.

As can be seen from this, the business analyst is sometimes an entry-level employee, sometimes a senior person, so be aware that the definitions used in this book are reflective of where the industry is heading, not necessarily where it is today.

If the analyst works in a large organization, assigned to a significant project, he or she may very well be a full-time business analyst. However, that is typically not the case for most business analysts. In addition to gathering requirements, they manage projects, design systems, build customer relationships, and may even make coffee in the morning. [Chapter 2](#) more closely defines what the current thoughts are on the role delineation of the business analyst. For now, though, keep in mind that when the term *business analyst* is used in this book, it is not describing a person, it is describing a role in the business. It may be 20 percent of a person's time, or it may take three people 100 percent (or more) of their time to fill the role ([Figure 1.2](#)).

The level of formality involved with the processes used by the business analyst will also vary greatly based on the tools used, the size of the organization, and the project, as well as the type of project that is being worked on. Naturally, if the project is a large, enterprise-wide initiative, the analyst will spend a lot of time analyzing the business, doing enterprise modeling, planning for organizational change, etc. However, if the



Figure 1.2 Business analyst role versus job.

project is to modify an existing report in a current system, there will likely be very little time spent on analyzing enterprise goals (or the customer will question the competency of the analyst).

This book is organized after the *BABOK*v3, with one chapter per knowledge area. It is important to note, however, that the topics in the book expand beyond the *BABOK*, and other relevant topics are introduced. It is also not intended to be used instead of the *BABOK*, so for those of you studying for certification, this book should be used together with the *BABOK*.

In addition to the IIBA, there are other organizations that have influenced the evolution of the business analysis profession, including the large toolmakers, training companies, and other standard-setting bodies such as the Institute of Electrical and Electronics Engineers, the Software Engineering Institute (SEI™), and others. One that deserves a bit of extra attention is the SEI. Started in the early 1980s, the SEI initially targeted the software industry and looked at defined processes needed to successfully run a software project. Today's SEI goes beyond just software and is looking at engineering practices, but a lot of their standards still have a software feel to them. They did, early on, place a special emphasis on requirements practices, and they did it before many other organizations had caught on to it. Some of SEI's definitions will be explored in [Chapter 2](#).

1.7 Where Is It All Going?

Although the IIBA is doing a great job developing the framework for business analysis, it is a moving target. The domain that the IIBA has taken on within the borders of business analysis is very large, a lot larger than the domain that the PMI took on with project management. Part of the reason for this is that the PMI chose to deal with generic project management, processes that should be applied to all projects. On the other hand, the IIBA appears to be going into a greater level of detail when exploring development methodologies, modeling techniques, quality assurance, and organizational involvement. As an example, the business analyst is involved pre-project, during the project, and post-project, and not only at a cursory level. The business analyst has significant roles and responsibilities in all three areas; the project manager, on the other hand, is primarily responsible during the project.

Business analysis is also more immature than most other domains. Systems development has been done for only about 40 to 50 years, and business analysis in this context even less than that. This may seem like a long time, but compared with manufacturing, construction, and other disciplines, it is new. This means that there likely will be significant changes in the body of knowledge for many years to come. That is okay; the basic concepts of lessons learned and process improvements apply here as well. As the industry is changing, the practitioners need to change with it. This is an

industry where tools, methods, and best practices are changing, and changing drastically. Just look at a topic such as agile development (which will be further explored later): 20 years ago, few people had heard of it and virtually no one had used it. Today, most organizations have incorporated some aspects of agile development approaches in their projects.

1.8 Book Project

To give examples of the different techniques and tools discussed throughout the book, a fictitious project, called the Prescription Interaction Project, will be used. The decision to make it a fictitious project rather than a real one is based on many reasons. First, any real project is likely to get into unique details for a specific industry, which, while interesting, is not the purpose of this book. Second, no project would use all the techniques and tools discussed in this book, meaning that even a real project would need some fabrication to accomplish this. Third, and most important, there is a need for the project to be simple and intuitive, so that the main concern is on the business analysis portions, not the business itself.

The Prescription Interaction Project is a project being done by a large drug store chain, C.V. Green. It is traditionally U.S.-based, but is slowly expanding into Asia and Europe. C.V. Green has a vision of minimizing negative interaction between different drugs prescribed to its customers, and wants to aggressively identify any potential drug interactions as early as possible. The company has been leading the effort to standardize drug interaction reporting and is working with a consortium to create a shared database where potential drug interactions can be identified, while still safeguarding the rights of the individual customers. That is the starting point and the framework for the examples found in the following chapters.

Separate from the book project theme, there is also a case study project. This is the creation of an inventory and ordering system for Swede-Mart, which will be used for most of the case study activities found at the end of each chapter. [Chapter 10](#) provides a detailed description of the Swede-Mart case study.

1.9 Summary

Business analysis is still a relatively new discipline. Although there are many emerging standards in the industry, to a large extent it is still up to each organization to pick the tools, processes, and definitions that work in its environment. This book uses the IIBA Body of Knowledge (*BABOK*) as the main reference for the framework of business analysis. The goal of this book is to get an organization, or an individual, started on the path toward a formalization of business analysis processes, and also to serve as a study guide for individuals seeking certification in business analysis.

Activity

What are the biggest blockers to successful requirements gathering in your organization(s)? Either brainstorm as a team or by yourself. Create a list of eight to ten factors.

Prioritize them based on the importance for project success and then select the top five. For those top five blockers, document one or two things that your organization can do to minimize the blockers and one or two actions that you as an individual can take to minimize them.

Discuss these in an upcoming team meeting to educate the organization and to start toward the path of a more repeatable process for requirements gathering.

2

LAYING THE FOUNDATION

In theory, there is no difference between theory and practice. But, in practice, there is.

Jan L.A. van de Snepscheut

This chapter reviews the standards that will be used as a foundation for the rest of the book. Although many organizations strive toward implementing standard processes, this is often not as easy to do as it sounds. There are many organizations defining standards, and these standards are often different from each other. How does one select the *best* standard? In this book, the selection of which standards to use was based on what seems to have the strongest following today, as well as what I believe will be the dominating standards in the future. In addition, because this book was written in the United States, the standards will be U.S. centric. Regardless of which standards are used in your organization, it is important that the whole organization has the same definitions and the same processes. If not, business analysis (and project management) will be almost impossible to do in an efficient manner.

2.1 Objectives

- Define a project life cycle.
- Define the purpose of a body of knowledge.
- Review standards from the International Institute of Business Analysis™ (IIBA™).
- Review standards from the Project Management Institute™ (PMI™).
- Review standards from the Software Engineering Institute (SEI).
- Compare and contrast the standards defined.

2.2 Overview

While this book considers the IIBA and the *Guide to the Business Analysis Body of Knowledge® (BABOK®)* as the primary sources for requirements gathering and product definition standards, numerous other organizations such as the PMI and SEI are also involved with defining standards in this domain, and will also be recognized and discussed in this chapter. It is vital for the understanding of business analysis, and the IIBA, to realize that there is not necessarily a consensus on what should be included

in the body of knowledge in business analysis. It is likely, over time, that the industry will move toward agreement on these standards. Even the PMI, despite having been around for a long time, does not have industry-wide agreement on what project management is. Over the years, though, it has been moving the industry much closer to a prevailing view.

In recent years, standards have also become more important for a different, and probably more powerful, reason. The customer demands it. More and more organizations are using standards such as the SEI's Capability Maturity Model® Integrated (CMMI), the PMI's Organizational Project Management Maturity Model (OPM3), and the International Organization for Standardization (ISO) certifications to assess an organization's process maturity. Although the intent of many of these efforts was toward internal process improvement, there has been more of an external drive behind them in the last few years. Many customers, especially in the government, will not even allow a vendor to bid on business unless they can show the proper certifications. This often causes an organization to invest heavily in attaining the certification, sometimes at the cost of adding unnecessary overhead. The demand for certification appears to be growing globally and will probably be one of the main drivers for standardization. Although process implementations and improvement efforts are desirable, they need to be done at a pace that the organization can absorb. In other words, if too much change is introduced too quickly into an organization to obtain some certification, then the certification often becomes the purpose of the effort rather than a help in improving product development. This tends to lead to any certification results being short lived and lost within a few years.

These standards, when implemented and followed, can greatly improve the success of any project, but only if they are used by skilled business analysts. This chapter also reviews some of the key skills required to make a business analyst successful.

2.3 Life Cycle Definitions

Most projects use two different life cycles to create an information technology (IT) product. The first life cycle, the project life cycle, deals with the standard project management process groups of initiating, planning, executing, monitoring and controlling, and close down. These processes are concerned with standard project issues such as budget, risk, schedule, and procurement. They are done for all projects in a fairly identical way regardless of what the organization's business is, and are typically controlled by the project manager.

The second life cycle is the product life cycle, often called the systems development life cycle (SDLC). This life cycle is unique to the product being developed; IT typically has life cycles different from those in the construction or pharmaceutical industries. This life cycle describes the product development activities, such as design, prototyping, and testing, that are done to develop a product. The ownership of these activities often resides with business analysts, systems analysts, or other subject-matter experts.

To complicate matters further, there is not only one standard SDLC in existence for IT projects, there are many. Some of the various SDLCs will be discussed in [Chapter 4](#); however, for the purpose of establishing a base for the material in this book, a standard, simple, old-fashioned waterfall life cycle like the one shown in [Figure 2.1](#) will be used for this book. When this book discusses analysis, it may be what other life cycles call definition, or requirements documentation. The intent of the phase is the same in all of the life cycles: to document the product.

It is easy to see why this is called the waterfall approach and why it is preferred from a management view. Everything is flowing logically from the beginning of the project to the end. The main deliverables from each phase in the SDLC, as well as the roles and responsibilities for those deliverables are shown in [Figure 2.2](#).

When matching up the project process groups with the product life cycle, it helps to view the project processes as iterative in nature. The processes will be performed for

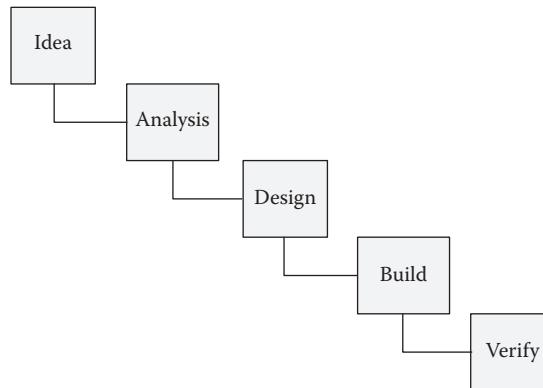


Figure 2.1 Standard life cycle.

Phase	Key deliverables	Roles and responsibility
Idea	Feasibility study Initial business case	Business owner
Analysis	Requirements document	Business analyst
Design	Specification	Systems analyst
Build	Product	Programmer
Verify	Test results	Quality assurance
	Sign off	Business analyst

Figure 2.2 SDLC phases, deliverables, and roles and responsibilities.

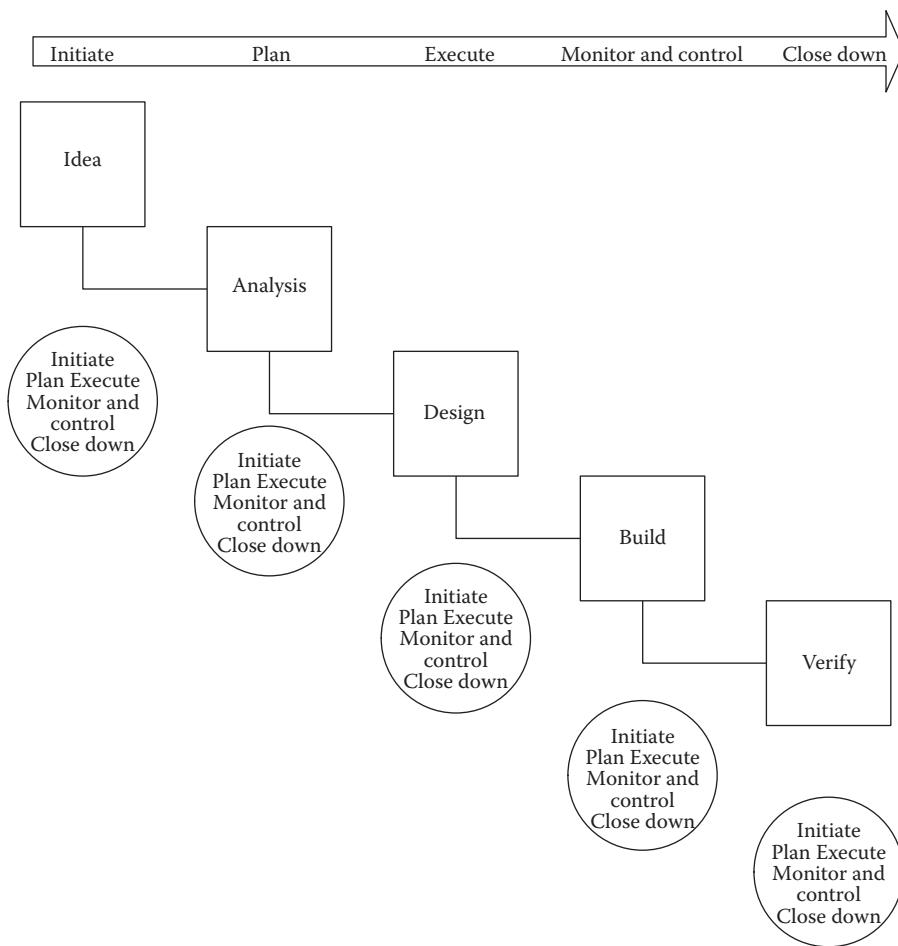


Figure 2.3 Standard life cycle and project management processes.

the whole project, and also for each individual phase of the project, which is shown in [Figure 2.3](#).

As shown in [Figure 2.3](#), while each project must be initiated, planned, executed, monitored and controlled, and closed down, the same is true for each phase of the project. Analysis, design, and other product development phases should be treated as their own dependent mini-projects with initiation, planning, execution, monitoring and controlling, and closing down. In addition, although the project manager oversees the project at a top level, frequently it is more efficient to have the business analyst or system analyst function as the sub-project manager for the individual SDLC phases.

2.4 What Is a Body of Knowledge?

A body of knowledge, be it that of the IIBA or PMI, strives to define all the knowledge that is pertinent within a profession, discipline, or specialty. That does not mean that every professional within the profession must be expert in all areas. It is desirable

for the professional to have an understanding of all areas but be an expert in only some of the areas.

It is important to understand the boundaries of the knowledge area being explored. For the PMI, that is project management; for the IIBA, it is business analysis. It reflects what is currently considered best practices in each respective domain. Note that both the PMI and IIBA publish a guide to each body of knowledge, which really provides an introduction to the topic, a definition of terms and boundaries, and an outline of key processes. The guide does not contain everything that the practitioner must know to be successful.

There are many unique specifics, variations, tools, and implementations for each organization. The guide shows the framework, the *what*. It is still up to each organization to define the *how*. An example of this is modeling. The *BABOK* discusses many types of modeling and also gives examples of detailed modeling techniques. That does not mean those are the only techniques to be used. Rather, it means that the business analyst should consider modeling requirements (both data and process), and that the techniques shown are some of the more common ones.

Some people have already been concerned that it may appear that the IIBA is promoting object-oriented modeling because it has a dominant role in the *BABOK*. That is probably more a result of the people writing those sections having more knowledge in those areas, and it should eventually have a more even distribution of what is used in the industry. Keep in mind that this is a moving target. The *BABOK* should reflect current practices. In an industry as dynamic as IT, current practices are likely to keep changing.

2.5 Overview of PMI Applicable Standards

The PMI's *A Guide to the Project Management Body of Knowledge*, Fifth Edition (*PMBOK™*), is currently viewed as the main standards document for project management. The *PMBOK* definition (on p. 3) of a project as "a temporary endeavor undertaken to create a unique product, service, or result" is today accepted by most project management organizations around the world.

With the ten knowledge areas and five process groups from the *PMBOK*, that book outlines what is, and in some cases what is not, project management. PMI's ten knowledge areas are

1. Integration management—handling the integration of the different components of the project plan. This is an overall knowledge area that overarches the other knowledge areas, making sure that they are well integrated. It also makes sure that the scope of the product is synchronized with the scope of the project. This knowledge area is where the project manager and the business analysts synchronize their efforts. Any change to the product or project scope by one will impact the other.

2. Scope management—defining the project scope, which means figuring out the work that needs to be done to deliver the desired product or service. The product scope or product description is an input, and the business owner (often through the business analyst) uses that to define the functions and features of the end product.
3. Cost management—estimating the effort and cost involved with creating the deliverables for the project. While the PMI does talk about life cycle costing and business cases, the *PMBOK Guide* covers mainly project budgeting.
4. Time management—determining the details and durations of activities to create a schedule for the project. This is the traditional project management area with Gantt charts and network diagrams.
5. Risk management—identifying, analyzing, responding to, and monitoring the project risk. Although product risk is also brought up here, the project manager is not typically the owner of product risks; that should reside with the business owner (again, often through the business analyst).
6. Human resource management—finding the resources for the project and building them into a productive team.
7. Procurement management—deciding on what to procure, going through the solicitation process, selecting a vendor, and then managing the vendor relationship through the project.
8. Communications management—defining the communications plan, and collecting and disseminating information and status reporting for the project.
9. Quality management—identifying the applicable quality standards for the project; determining what activity to do for quality inspection and quality assurance.
10. Stakeholder management—identifying stakeholders, and managing and controlling the engagement with those stakeholders.

It is important to note that although these are the key project management-related knowledge areas, they do not represent all the skills needed by a project manager to be successful. The project manager must also possess general management skills, application area skills, and leadership skills. However, the knowledge areas highlight the unique project manager skills for a project. Although not all projects will have a procurement portion, the other nine knowledge areas would be represented on all projects to some extent.

The PMBOK also shows the five process groups that any project must go through. How they are done may vary, but they must be done at some level of detail:

1. Initiating—starting the project or phase and getting buy-in to the activities
2. Planning—creating a plan for how to create the project or phase deliverables
3. Executing—performing the actual work outlined in the product development life cycle

4. Monitoring and controlling—evaluating and comparing actual performance to planned performance and taking corrective actions when needed
5. Closing down—finishing the project or phase and getting acceptance of the deliverables

The output from the work of the business analyst, the business requirements document, is actually what the PMI refers to as product description and is largely an input into the project management processes.

On many projects, the business analyst also becomes a project manager for the analysis phase of the project; thus, for the project management portion of the analysis phase, many of the tools and techniques defined by the PMI can be used. The knowledge areas of scope management, time management, human resource management, quality management, risk management, and communications management are especially relevant. [Chapter 4](#) shows how those standards can be adapted for a typical (if there is such a thing) business analysis project.

Project management and business analysis are disciplines that tend to interact heavily. Although they are clearly distinct domains, it is important for the practitioners in one area to be aware of deliverables and processes in the other.

On many projects, the business analyst also becomes a project manager for the analysis phase of the project; thus, for the project management portion of the analysis phase, many of the tools and techniques defined by the PMI can be used.

PMI has rolled out its OPM3 as a way of measuring an organization's project management maturity. Although not a major force in the industry yet, it is likely to grow in acceptance as PMI improves and enhances it.

2.6 Overview of IIBA Framework and Standards

The current version of the IIBA's *BABOK* is version 3. It defines business analysis as "... the practice of enabling change in an enterprise by defining needs and recommending solutions that deliver value to stakeholders."

When it comes to defining what a requirement is, the IIBA initially borrowed from the Institute of Electrical and Electronic Engineers Standard 610.12-1990, but enhanced it to better suit today's environment. In the IIBA definition, a requirement was, up until version 2 of the *BABOK*,

1. A condition or capability needed by a stakeholder to solve a problem or achieve an objective
2. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document
3. A documented representation of a condition or capability as in 1 or 2

Dissecting these statements makes it a bit clearer. A *requirement* is something that is needed to solve a business problem (as in 1) or to deal with an outside constraint such as a corporate standard or a government regulation. Item 3 adds that it must be written down. It does not count as a requirement if it is just a mental note. It has to be documented for traceability and for change control.

In *BABOK* v3, the IIBA changed and simplified this definition to “a usable representation of a need.” This is a more general statement but also one that may be easier to relate to for all stakeholders.

The IIBA has identified six knowledge areas that are all a part of the *BABOK*, and each of these will be expanded on in their own chapters:

1. Business analysis planning and monitoring: This deals with the planning of the requirements-gathering activities, analyzing stakeholders, and getting buy-in to the requirements-gathering efforts. It also outlines how change control and traceability of requirements will be dealt with.
2. Elicitation and collaboration: Different projects need different approaches to capture requirements. Sometimes, it may be a survey, sometimes an interview. This section documents best practices and identifies pros and cons with each approach. It is the job of the business analyst to select the best technique for each stakeholder on each project. It also deals with the ongoing communication with stakeholders after the requirements are elicited.
3. Requirements of life cycle management: This looks at how we maintain the requirements and design information through the project phases.
4. Strategy analysis: Business analysts must understand the business and organizational environment in which they are working and recognize that a project does not operate in a vacuum, but rather is interacting with the rest of the enterprise. For the business analyst to successfully document requirements, the business, its goals, visions, business rules, and mission must be documented as well. This knowledge area also deals with feasibility studies and evaluation of business cases for different projects.
5. Requirements analysis and design definition: After requirements have been elicited, they must be organized, analyzed, and documented. This may be in text format or in models, or in a combination of both. Conflicting requirements should be identified, and the set of requirements should be evaluated for priority and completeness. It is also here that alternative solution options are identified.
6. Solution evaluation: The business analyst must trace the requirements through the product cycle and ensure that the product being designed and developed matches the requirements given by the stakeholders. This includes verifying that the customer is satisfied with the product after implementation.

In addition, the *BABOK* has chapters on key concepts, underlying competencies, perspectives, and techniques. The first three of these will be covered in the *BABOK*

overview chapter. The techniques will be covered within the knowledge areas with examples and discussion on how and when to use them.

Business analysts must understand the business and organizational environment in which they are working and recognize that a project does not operate in a vacuum, but rather is interacting with the rest of the enterprise.

The IIBA standards are among the most recent additions to the field of standards. That probably means that they are likely to change the most. At this time, they have been developed by a group of dedicated and competent people receiving some feedback from the community. It is likely that as the exposure of the standard increases, there will be more diverging views on what should and should not be a part of the *BABOK*. Over the next few years, there will likely be more changes and new versions of and additions to this standard versus the others reviewed in this chapter. So does that mean that an organization would be better off waiting until the standard is better established? There are both positives and negatives with either approach.

If an organization waits until a standard is fully developed, it runs the risk of losing years of working with a very good, albeit not perfect, standard, as well as losing the chance of impacting the direction of that standard. By getting on board early, the organization is more likely to be viewed as innovative and an industry leader. However, the organization does need to be flexible and able to deal with change. It is also worth looking at what the expectations are from the organization investing in the standards. If it is to learn best practices and a standard approach to business analysis, the organization will need patience; if the expectation is to provide a competitive advantage by moving staff toward certification, then the earlier the effort to get up to speed on IIBA concepts begins, the more benefit there will be.

2.7 SEI-CMMI and Applicability

In the early 1980s, the Department of Defense (DoD) was looking for a way to evaluate systems engineering vendors to see which ones had mature development processes and, as a result, were more likely to meet their time and cost estimates with good quality. The DoD funded the SEI, located at Carnegie Mellon University in Pennsylvania.

The SEI has defined a requirement as “something that the product must do or a quality that the product must have.” This definition is much simpler to understand than the IIBA definition and, as such, is probably more usable in real life. This definition can be shared with the customer and easily understood.

The SEI developed a maturity model evaluating the capability of an organization to implement repeatable development processes. The current version is called CMMI version 1.2. It identifies five levels of maturity for an organization, from ad hoc fly-by-night to being totally obsessed with process improvement. It identifies a number of

activities that need to be performed to reach each level. Two of the key activities identified, as they relate to business analysis, are requirements management and requirements development. Every key activity in CMMI is associated with goals for the area and key practices for its successful implementation.

2.7.1 Requirements Management

The SEI states that “The purpose of requirements management is to manage the requirements of the project’s products and product components and to identify inconsistencies between those requirements and the project’s plans and work products” (<http://www.sei.cmu.edu>).

This implies a concentration on managing the requirements after they have been identified and documented. It is interesting to note that the SEI actually sequences this activity before the actual activity of capturing requirements from a process viewpoint. The thought is that unless there is a formal approach to getting buy-in and reviewing the requirements with key stakeholders, there is not much point in formalizing what the process should be for capturing those requirements. The approach makes sense; however, most organizations appear to spend much more effort on eliciting the requirements than on validating them.

The goals and key practices for the requirements management area are

Goal 1: Manage requirements

- 1.1 Obtain an understanding of requirements.
- 1.2 Obtain commitment to requirements.
- 1.3 Manage requirements changes.
- 1.4 Maintain bidirectional traceability of requirements.
- 1.5 Identify inconsistencies between project work and requirements.

Goal 2: Institutionalize a managed process

- 2.1 Establish an organizational policy.
- 2.2 Plan the process.
- 2.3 Provide resources.
- 2.4 Assign responsibility.
- 2.5 Train people.
- 2.6 Manage configurations.
- 2.7 Identify and involve relevant stakeholders.
- 2.8 Monitor and control the process.
- 2.9 Objectively evaluate adherence.
- 2.10 Review status with higher-level management.

At first glance, these practices appear almost obvious: understand requirements, manage changes to requirements, and establish traceability. However, these are the areas that tend to cause projects to fail, largely because these processes are not performed consistently and with discipline within many organizations. The focus here is

on repeatability and consistency. If an organization establishes traceability among the business need, the requirements, and the product; gets organizational commitment to the requirements; and handles changes in the requirements in an organized approach, many of today's project frustrations would be eliminated.

2.7.2 Requirements Development

The SEI states, "The purpose of requirements development is to produce and analyze customer, product, and product–component requirements." This is where the requirements are actually captured and analyzed. Once that is done, the requirements can be managed and controlled because of the processes implemented earlier in requirements management.

Goal 1: Develop customer requirements

- 1.1 Elicit needs.
- 1.2 Develop the customer requirements.

Goal 2: Develop product requirements

- 2.1 Establish product and product–component requirements.
- 2.2 Allocate product–component requirements.
- 2.3 Identify interface requirements.

Goal 3: Analyze and validate requirements

- 3.1 Establish operational concepts and scenarios.
- 3.2 Establish a definition of required functionality.
- 3.3 Analyze requirements.
- 3.4 Analyze requirements to achieve balance.
- 3.5 Validate requirements with comprehensive methods.

Goal 4: Institutionalize a defined process

- 4.1 Establish an organizational policy.
- 4.2 Establish a defined process.
- 4.3 Plan the process.
- 4.4 Provide resources.

These goals and practices are more comprehensive than those for requirements management. This is partly because requirements development emphasizes best practices much more and is a much larger domain than requirements management. This also correlates to the different CMMI levels. Level 2, where requirements management resides, is about project management and discipline. Level 3, where requirements development is, deals more with systems engineering and defined processes.

2.8 Which Standard to Use?

All the standards described have different specialties and different strengths. In addition to those discussed here, numerous other excellent standards have been developed,

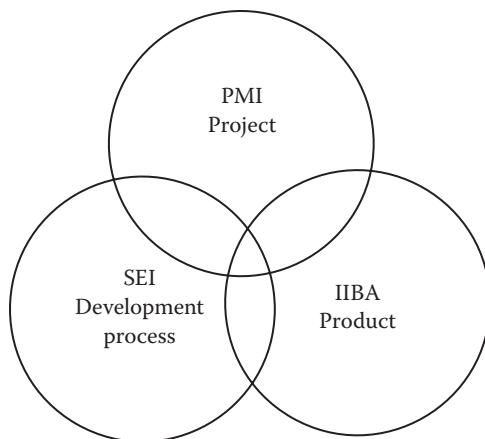


Figure 2.4 Standards and their strengths.

such as Six Sigma and Total Quality Management. It is more important that the organization uses a standard and uses it consistently than the specific standard the organization has selected to use. In the IT industry in general, there tends to be a focus on three dimensions: the project, the product, and the development process. [Figure 2.4](#) shows these three dimensions and the main focus for each standard. So what does that mean? Should an organization pick all three? Or just one?

One good approach is to select a foundational standard. If the primary concern of the organization is to provide training on requirements gathering, then the IIBA can serve as a foundation. If the goal is to implement solid processes for software development, then SEI-CMMI may be the best, keeping in mind that it does require a significant organizational commitment. If the main issue is getting the job done, resource planning, scheduling, and budgeting, then the PMI can help with the groundwork. While learning from all, it is best to start with one; otherwise, there can be confusion and lack of buy-in from the organization. Probably the most commonly heard complaint about implementing a new process initiative is that it is new, inconsistent with other standards, and adds overhead and bureaucracy. Although that is often true, its impact can be significantly less if the organization picks one standard and then stays with it. The main benefit from adopting standards is down the road, when they become second nature. So be patient and do not move from the SEI to the PMI to the IIBA (with some Six Sigma thrown in) within a short time. Also, if the organization is adopting multiple standards, then create a map that shows the intended use of each standard and how they interact with each other.

2.9 Comments on Tool Standards

As a part of the job of the business analyst, it is likely that some (or all) of the requirements will be documented in some package or tool. This can be as simple as SmartDraw™ Visio™, or something more sophisticated such as Oracle™ Designer or

Rational. All of these packages have standards built into them, and the more sophisticated the package, the more rigid the standards. It does make sense to attempt to follow these standards as closely as possible. Trying to circumvent the standards in a package tends to lead to extensive extra work and a result that is less than optimal. Most tools try to be consistent with the PMI and other such standards, but the tools do not always keep in sync with changes to the standards. When looking at what tools to use, review what expertise the organization already has. It is much more important that a tool is selected where the expertise already resides within the project team than to pick a *neater* tool that no one knows how to use.

Activity

Take inventory of what standards, certifications, and processes exist in your organization. These standards should include what training is being done, the forms and templates that exist in the organization, and the tools being used. Even if there is no book sitting somewhere called *Standards*, by asking around and talking to experienced practitioners a base of tools can be discovered.

Document them in the following table:

ORGANIZATION	STANDARD	PROCESS OWNER	IN USE SINCE



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

3

BABOK OVERVIEW

The road to success is always under construction.

Steve Maraboli
Life, the Truth, and Being Free

This chapter presents the *Guide to the Business Analysis Body of Knowledge® (BABOK® v3)*. It will discuss some of the general sections such as key concepts, underlying competencies, and perspectives.

3.1 Objectives

- Identify key concepts from the *BABOK*.
- Identify key skills needed by a business analyst.
- Discuss different perspectives on business analysis.

3.2 Overview

The core sections of the *BABOK* are the knowledge areas and the techniques. However, in addition, the *BABOK* also contains some more overarching sections, with impact on all business analysis work. There are some general concepts, including the Business Analysis Core Concept Module™ (BACCM™). There are also the different views, or perspectives, that different types of initiatives will have on the business analysis activities. Finally, there is a need to look at the key skills and competencies needed by a business analyst.

3.3 Key Concepts

In *BABOK* v3, the IIBA introduces the BACCM. It gives a basic foundation for the whole *BABOK* with common concepts and common definitions ([Figure 3.1](#)).

The interrelated concepts from the model are

- Changes
- Needs
- Stakeholders
- Value
- Solutions
- Context

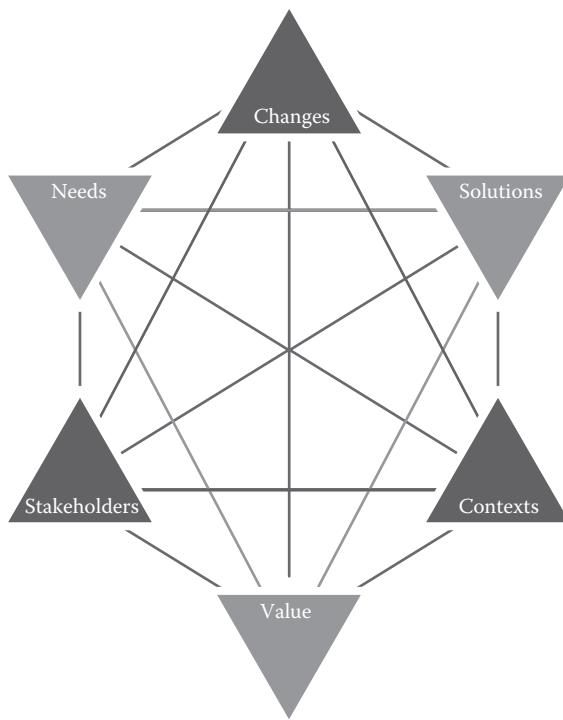


Figure 3.1 Business Analysis Core Concept Module. (© 2015 International Institute of Business Analysis™. All rights reserved.)

These concepts can be viewed as an aid to understanding business analysis initiative. By ensuring understanding the concept and the relationships between the concepts, the business analyst can evaluate what actions are needed and the best approach for the initiative.

3.4 Perspectives

Business analysis is such a broad and varied discipline, so the process of collecting tools, techniques, and best practices can sometimes be confusing. To shed some light on this, version 3 of the IIBA *BABOK* added a chapter on perspectives. Five perspectives were selected:

- Agile
- Business intelligence
- Information technology (IT)
- Business architecture
- Business process management

This is by no means an exhaustive list, but it reflects some of the most common views on business analysis work. It also important to note that many initiatives will

contain multiple perspectives. The *BABOK* v3 expands greatly on each of the perspectives. Here, I would like to discuss what the main unique drivers are for those perspectives.

3.4.1 Agile

Agile development has been one of the most discussed new trends in product development and has often been misunderstood. Agile is maybe more of a mindset than a technique. There are many agile methodologies, but they have a common ground in that the expectation of change is built into the methodology. This requires a willingness to learn, to rethink, and to be flexible. This is very different than the old days of massive preplanned undertakings where changes were discouraged and expensive. Techniques for agile initiatives are used to enhance communication and are often quite simple since change is expected.

3.4.2 Business Intelligence

When one first looks at business intelligence, it is easy to get the feeling that this is as far away from agile as you can get. This is about data analysis. The techniques used here would be very different: data modeling, decomposition, metrics, key performance indicators, and such. And while that is true, it is easy to envision an initiative where we would use an agile view and a business intelligence view; for example, an executive information system, where we would develop the interface and requirements using an agile mindset and then focus on business intelligence when we create the underlying platform.

3.4.3 Information Technology

Many business initiatives are constrained by, and have an impact on, IT. This can be as varied as the need for integration to existing software and the need for an enhanced or redesigned information technology infrastructure. Interface analysis, process modeling, and workload simulations may be a part of understanding the need. Most initiatives would have some need to consider the IT view.

3.4.4 Business Architecture

This involves looking at modeling the enterprise. To understand the current capabilities of the business and to set the vision of where the business is heading, there is a need to understand goals, competencies, capabilities, and resources. This is often quite a different business analysis activity, with senior stakeholders and great complexity. The output of these initiatives will create a foundation for most other initiatives in an organization.

3.4.5 Business Process Management

A very common view of new initiatives is the process view. Which business processes are involved, impacted, and in need of improvement. The business analyst role can here work closely with process architects, analysts, and modelers. Simulations, value analysis, and quality evaluations may be part of these initiatives.

3.5 Business Analyst's Skills

One of the areas in the *BABOK* that needs the most augmentation right now is the underlying fundamentals. The current version of the *BABOK* contains a list of some business analysis skills; however, there is no elaboration or detail to describe the skills. Although many of them are fairly obvious, it is important to look at them in the light of business analysis rather than just regular soft skills. Some of the key skills from the *BABOK* list are outlined here.

3.5.1 Analytical Thinking and Problem Solving

This may seem too obvious to be stated because it is built into the name of the profession; however, the business analyst is not just a documenter or an interviewer. The business analyst needs to be able to organize and analyze a variety of information for conflicts, impacts, and priorities. This includes structured analysis, root-cause analysis, and impact analysis.

Structured analysis techniques, such as the ones covered in [Chapter 8](#) of this book, allow the business analyst to take abstract concepts, complex scenarios, and disjointed customer wishes, and put them all into a structured document that can be used by the developers as a basis for developing a system.

Root-cause analysis is another skill at the core of the business analysis profession. The customer will often bring problems, or sometimes even just symptoms of problems, to the business analyst and expect a solution. It is up to the business analyst to work with the customer to figure out what the root cause is. A key help in root-cause analysis is to ask why. Why is this a problem? Why is this happening? Any solution that does not address the root cause is likely to be incomplete and short lived.

Finally, what is impact analysis? The customer must understand the impact of introducing the solution to a problem into an organization. Although customers normally understand the benefit of getting rid of the problem they have, they may not be clear on the impact of the solution. Looking at the Prescription Interaction Project from [Chapter 1](#), the customer likes the thought of being able to identify and stop dangerous drug interaction from prescriptions used by different family members; however, the customer may not like the potential privacy breeches that may result from sharing these data among family members. In this case, the customer may be trading one set of lawsuits for another set. The business analyst must evaluate and bring those types of issues to the customer's attention.

3.5.2 Behavioral Characteristics

For the business analyst to build trust, they need to show integrity and a strong sense of ethics, and take responsibility for their actions. Managing their time well and being structured is important while at the same time being adaptable to the needs of the surrounding.

3.5.3 Business Knowledge

The business analyst must understand the business of the customer, the marketplace, the products, and the competitive position of the customer. The analyst must also understand the business processes and the systems that currently support the business. Although business analysts can come from the IT side or from the business side, it does appear that more of them are from the business side today than was the case only 10 years ago. There may be many reasons for that; my personal view is that there are two primary drivers:

1. Knowing and understanding the business can be harder than understanding the IT area. It is often easier to teach a savvy business person enough about systems to be successful than to teach a systems person about the business side. Although this certainly depends on the environment, usually the business drives technology and not the other way around. There are exceptions, such as a global data-mining operation, where technology is actually driving and changing the business.
2. The business is realizing the importance of IT solutions and is no longer comfortable delegating to the IT department the responsibility of determining how to best take advantage of systems.

Although business analysts can come from the IT side or from the business side, it does appear that more of them are from the business side today than was the case only 10 years ago.

Regardless of the background of the analyst, he or she must receive continual training and exposure to the business environment.

3.5.4 Communication Skills

A person can be great at regular conversation, but fail as a business analyst. To get requirements from a customer, the business analyst must provide a structure and a flow to the discussion during the collection of information. A conversation may garner significant information, but it is likely to have holes and lack detail. An interview will be structured to go from high level to details, from general to specific. It will provide a systems way of thinking and a logical flow to the session. Requirements classifications, covered in [Chapter 8](#), help with this by starting at the top level, understanding

the business needs, and gradually decomposing that understanding into user requirements and eventually into system requirements. The business analyst must facilitate this flow by preparing and asking the right questions at the right time.

This is a skill that can be learned and improved; however, for some people, this type of systems thinking comes more naturally. Those people are good business analyst candidates in any organization.

Although business analysts are not necessarily managers of the tasks performed, they must be able to lead a diverse group of people through the analysis phase. As leaders, they must provide the context of the activities—in other words, why the activities are being done. If the participants do not see the value of root-cause analysis or process modeling, then it is unlikely that they will put their best efforts into them. And, in many cases, business analysts will lead a group of other analysts, developers, and users in capturing requirements and building a consensus for a solution.

3.5.5 Interaction Skills

The business analyst's job, unlike the programmer's, is not a solitary one. Most of the business analyst's time will be spent interviewing, meeting, and presenting to stakeholders. Solid time management skills, the ability to stay on target, and the ability to logically guide a group of people through elicitation, validation, and buy-in is key for the business analyst.

Most of the presentations and meetings that the business analyst will facilitate will be to discuss or review information that comes from other stakeholders. It takes a certain talent to be able to present this in a neutral and unbiased manner, while still being enthusiastic about the project. The business analyst's mindset should always be "Here's what the stakeholders have decided" rather than "Here's what I have decided."

Although this skill can be enhanced by training and practice, some of it is based on personality as well. The business analyst profession is like many other professions: a person with the right personality and characteristics is much more likely to be successful.

Business analysts are not traditionally viewed as decision makers, at least not regarding the product. However, they must be decisive when it comes to the process, the meetings, the commitment, and getting stakeholder buy-in to both process and product. The ability to effectively accomplish these tasks requires strong negotiation skills. The analysis phase of the project is full of negotiation opportunities. There will be negotiation about time and resources for the analysis activity, about which stakeholders can be interviewed, about the approach to gathering requirements and the best way to document those requirements, and much more. The analyst must be aggressive as far as getting what is needed for the work, but also realistic. There are competing priorities in all organizations. Make sure to always keep the project objectives in the forefront of any negotiation. Ask: "How will the outcome of this negotiation impact the project's abilities to meet the objectives?"

Conflict resolution has a number of applications for the business analysts. There can be conflicts with the customers, with other analysts, and with developers, just to name a few. Part of conflict resolution is also to know when to escalate. Business analysts must understand the organizational culture and know the individuals well enough to be able to decide when to solve the conflict versus when to escalate it. This escalation can be to the project manager, to the customer, or to the sponsor. Normally, though, the business analyst should work through the project manager with most issues needing escalation.

3.5.6 Tools and Technology

Although the mindset should be on *what* and not *how*, business analysts cannot be successful unless they understand the systems and technologies that are supporting the business. They may not need to be able to write a computer program, but they must understand what is needed by the developers to write a good program. Difficulties in communicating with the developers can usually be traced to a poorly defined business requirements document (BRD). The content and level of detail included in the BRD must be predefined, and this activity, assuming that the project will be an in-house development, should be driven by the development team. The development team is the primary audience of the document, and if it does not provide enough information for the team to use for development, then the effort has failed. So make sure that the BRD template is developed as a joint effort by business analysts and developers, where the balance of power resides with the developers.

3.6 Summary

No organization can be successful in the long run without a common language and an ability to move the entire organization in the same direction. That is the purpose of implementing standards. It allows the participants on a project to know what is expected of them, to identify standard deliverables, and to ensure that lessons learned from past projects are implemented on future endeavors. With all the standards available, it is more and more common that different parts of an organization are adopting different standards. This can cause confusion throughout the organization and often leads to a feeling of each initiative being the *flavor of the month*, rather than something that has full management commitment behind it. However, the standards will not substitute for skilled business analysts. A skilled business analyst will get better with good processes. An unskilled business analyst will struggle with or without processes.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

4

BUSINESS ANALYSIS PLANNING AND MONITORING

In preparing for battle I have always found that plans are useless, but planning is indispensable.

Dwight D. Eisenhower

Project management is a more mature discipline than business analysis, and the standards established for project management have been well defined by the Project Management Institute® (PMI®). There is nothing contradictory to those standards in this chapter; project management for a business analyst is conceptually the same as any other type of project management. With that said, what this chapter emphasizes are the areas of project management that are most likely to be of interest to a business analyst; the chapter also looks at some of the unique challenges that the business analyst may face during the analysis phase of a project.

4.1 Objectives

- Review the tasks identified by the International Institute of Business Analysis™ (IIBA™) for business analysis planning and monitoring.
- Review the role of the business analyst in project planning.
- Identify the components that may be a part of a requirements plan.
- Provide a template for a requirements plan.
- Review project management practices within the context of the analysis phase.
- Understand the importance of stakeholder analysis.
- Create a requirements plan for a sample project.

4.2 Overview

The business analyst is typically responsible for the planning of the requirements-gathering effort. Although the extent of these responsibilities varies greatly from organization to organization, there is a trend toward having the person with the most knowledge about the activities performed also being the person planning them. For the analysis phase of the project, that person is the business analyst, and for the design phase, it may be the systems analyst. It is important to note that the responsibility of

this planning effort is delegated from the project manager, meaning that approval, status reporting, and escalation should typically be done through the project manager, rather than directly with the customer, sponsor, or other stakeholders. Therefore, from a planning standpoint, the business analyst should regard the project manager as the customer. This is an important distinction. If the business analyst is not clear on exactly what authority he or she has, and how to communicate with the project manager and the customer, it is likely that there will be some conflict and frustration during the project.

If there are a number of business analysts assigned to the project, one should be designated as the coordinator or planner. That person functions as a team leader for this phase, coordinates and assigns tasks, and interfaces with the project manager from a management standpoint. The other business analysts may still be planning their individual activities, but there should be a lead analyst coordinating the whole analysis phase. Of course, on many projects, especially small ones, the project manager may retain all planning responsibilities, meaning that he or she actually takes on the analyst's planning role as well as the overall project-planning role.

In the survey of business analysts mentioned in [Chapter 1](#), it was clear that the biggest difficulties in gathering requirements was lack of process focus, lack of time, and lack of buy-in to scope. All of these areas can be enhanced by good planning practices.

4.3 IIBA View

The IIBA has identified five tasks in this knowledge area:

- Plan the business analysis approach.
- Plan stakeholder engagement.
- Plan business analysis governance.
- Plan business analysis information management.
- Plan business analysis performance improvements.

4.3.1 *Plan the Business Analysis Approach*

I often ask this question in my classes: Why do you plan? It seems like such an obvious activity that the participants often struggle to come up with meaningful answers. Common answers include

- To create a schedule
- To find out what it will cost
- So we know what to do
- To give us a baseline

These are all correct answers; however, the two answers that most often are missed are stakeholder buy-in and negotiation. And these two may be the most important for the success of the project.

1. Stakeholder buy-in: For a business analyst, getting access to users and other stakeholders is vital. If the customer is invited to a 2-day requirements-gathering session and is given only 1 day's notice, it is unlikely that the customer will send the best people to it. However, if there is a well-documented plan and the customer has 3 to 4 weeks' notice before the session, the likelihood of customer commitment is greatly increased. So, plan early and involve the stakeholders in the planning process. Many projects fail because of poor requirements, and one of the root causes of poor requirements is often that the right people were not involved. Lack of customer availability is the number one difficulty mentioned when gathering requirements.
2. Negotiation: Most (if not all) projects have resource constraints. There will be pressure to cut costs and time at each phase of the project. If the estimate for the analysis phase is based on gut feeling and intuition, it is unlikely that the decision makers will be convinced that all the time and money asked for is truly needed. However, if there is a detailed plan with well-defined tasks and deliverables, the discussion will focus on which tasks should or should not be done, rather than the quality of the estimate given for the project. In negotiation, the person with the most information will always have a clear advantage.

A plan is a stake in the ground against which progress is measured. There is no such thing as a perfect plan; at least not once the project is underway. The focus should be on creating a plan that all parties are comfortable with, and then deal with any variance to the plan through corrective actions. So decide how much time and effort should be spent on planning for each specific project. If the business analyst and the customer are co-located, and both parties are experienced and understand both the business and the systems environment, then the planning effort can be fairly minimal. But there still must be a plan.

If the business analyst and the customer are co-located, and both parties are experienced and understand both the business and the systems environment, then the planning effort can be fairly minimal.

Manage the plan whenever possible. After the project has started, when there is a need to update stakeholders on status, that update will largely consist of documenting actual progress and comparing it to the planned progress. If the discrepancy between the plan and the actual result gets too large (what project managers call a significant variance), there may be a need to re-baseline, meaning re-plan the project. This is not a preferred approach because it tends to lessen accountability; however, it is preferable to

following a plan that is widely considered to be unreachable. However, the first choice should always be to take corrective action to bring the project in line with the plan. There is often a significant organizational commitment to a plan, and if the plan keeps changing, the organizational commitment is likely to wane.

4.3.1.1 Overview of Approaches Software development, and requirements gathering as a subset of it, is still a relatively immature industry. This means that processes, frameworks, tools, and techniques are still evolving rapidly. In the early days of software development, there were no processes in place. Products were simple and the entrepreneurs who developed them often worked by the trial-and-error method. There was nothing wrong with this approach. On small projects, it may still make sense to try to develop something quickly, show it to the customer, get feedback, and then rewrite it if needed. Some people actually call this methodology *code and fix*, and elements of this approach are actually popular again in what today is called *agile* development. However, as the complexity of the product as well as the number of customers involved increases, it will soon become clear that some structure is needed.

From traditional waterfall life cycles in the 1970s, through iterative and spiral development in the 1980s, software development has now evolved to adaptive and agile life cycles for today's projects. Some of these concepts are mostly buzzwords and are implemented without a true understanding of the steps involved; however, most of them can add value to the development process. The most common life cycles are waterfall, iterative, and agile, which are explained below along with their advantages and disadvantages.

The continuous creation of new development approaches, tools, and techniques makes the software development domain seem like a moving target. Part of this is due to the immaturity of the industry and part of it is due to the great differences in the types of projects to which these approaches are being applied. It is easy to see that the process and tools used to build a tree house in the backyard may not be the same process and tools used to build a skyscraper in a large city. But for software developers, there is often an attempt to fit the same tools, the same process, and the same approach to very small projects as to large enterprise efforts. As the PMI states in the *Guide to the Project Management Institute's Body of Knowledge*, Third Edition (*PMBOK*), realizing that all projects are unique is a good starting point.

Most projects are a mix of multiple life cycles where some phases are run with a waterfall approach, some iterative, and often some with attempts at agility thrown in. It is important to understand that the type of life cycle used largely provides a mental picture for how the project is run. There is nothing that prevents the project from doing iterations in a waterfall life cycle or from applying agile techniques in either of them. The methodology gives an image of how the project will be managed. It tells the stakeholders what level of formality is planned; it shows if this project is an exploratory endeavor or if it is predetermined. In short, it puts the team on the same page.

Most projects are a mix of multiple life cycles where some phases are run with a waterfall approach, some iterative, and often some with attempts at agility thrown in.

The IIBA has simplified the different development approaches into plan driven and change driven. A plan-driven approach assumes that most requirements can be determined up front and that the customer has a clear understanding of what it wants. In a situation like this, there is less of a need for iterations, prototyping, and frequent checkpoints. A change-driven approach, on the other hand, assumes that the product requirements will evolve throughout the development. In these situations, a more agile approach is needed. Below is a more in-depth view of processes and methodologies for the undertaking of software development projects.

4.3.1.2 Selecting and Customizing a Process Good business analysts (just like good project managers) have a toolkit available to them. In this toolkit, they will have processes that can be used in various situations. Some projects may require just a sit-down interview with the customer; others may need to mix in some iterative requirements sessions of gradually increasing levels of requirements detail. Yet others may work best by sitting next to the customer and working out the requirements together in a more dynamic environment. The point is that each situation is unique. The business analyst who always uses the same approach and the same techniques will become ineffective. What worked the last time may not work on the next project. So when starting a new project, review the methodologies discussed in this chapter, add in your own experiences, and then decide what to do for the current project.

The Software Engineering Institute (SEI), for level 3 of the Capability Maturity Model® Integrated (CMMI) certification, has a knowledge area called *integrated project management*. The SEI describes this knowledge area as follows in their publication CMU/SEI-2002-TR-012:

The purpose of integrated project management is to establish and manage the project and the involvement of the relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes. For integrated product and process development, integrated project management also covers the establishment of a shared vision for the project and a team structure for integrated teams that will carry out the objectives of the project.

So what does that mean? To start with, there are three main components here. First, an organization should have a set of processes and approaches that it uses for different types of projects. These processes may be internally developed, they may come out of a book, or they may be associated with a specific tool. This organizational library of processes (as well as tools and techniques) is necessary to be able to

implement continuous process improvement in an organization. After each project, lessons learned should be added to the process library. This will minimize the likelihood of repeating the same mistakes.

Second, a stakeholder analysis and coordination effort is undertaken. Stakeholder analysis is done for many reasons during a project, and different views of that are shared throughout this book. At this time, the focus is to understand what type of project this is from the stakeholders' view. If their view is that this is a global, enterprise-wide undertaking where the company's future success is at stake, then a formal approach will be needed. However, if they view this as a discovery process, blazing the way for new approaches, then a more flexible, or agile, approach would be warranted.

Third, the project's vision is defined, understood, and communicated to all stakeholders. This includes an understanding of what the project itself is all about as well as identifying interfaces, constraints, and critical success factors. Without this being established, half the organization may view this as a research and development project, the other half as an operational improvement.

By the way, these steps are not done sequentially; there is really one effort with three focus areas that will impact each other. When all three are understood, then the customized process for this specific project can be established. This process may exactly follow one of the organizational processes, or it may be a combination of a number of them; however, this is where the project manager at the project level and the business analyst at the requirements level define what approach their efforts will take.

The book will cover some specific approaches below but, in sum, the requirements phase can have two general approaches:

Formal—for projects that are large and complex or for projects with stakeholders spread over many locations and where consensus building is needed. This approach is also good for when the customers have a good understanding of what they want and there is a desire to avoid scope creep.

Flexible—for projects where the end result is not yet well understood. It works best in an environment where stakeholders are co-located and where the projects are either small or can be decomposed into smaller, well-defined subprojects.

4.3.1.3 Waterfall The waterfall life cycle at the project level was introduced in [Figure 2.1](#). In this chapter, the attention will be on the analysis phase and the waterfall approach as it applies to this phase. [Figure 4.1](#) shows the approach of going through the analysis phase one step at a time, and finalizing each step before moving on to the next step. This is the key premise of the waterfall.

Step 1 is to understand the project and product scope. This means establishing how the project fits inside of the enterprise, what strategic goals it will help accomplish, and what dependent projects there are in the organization. A large portion of this

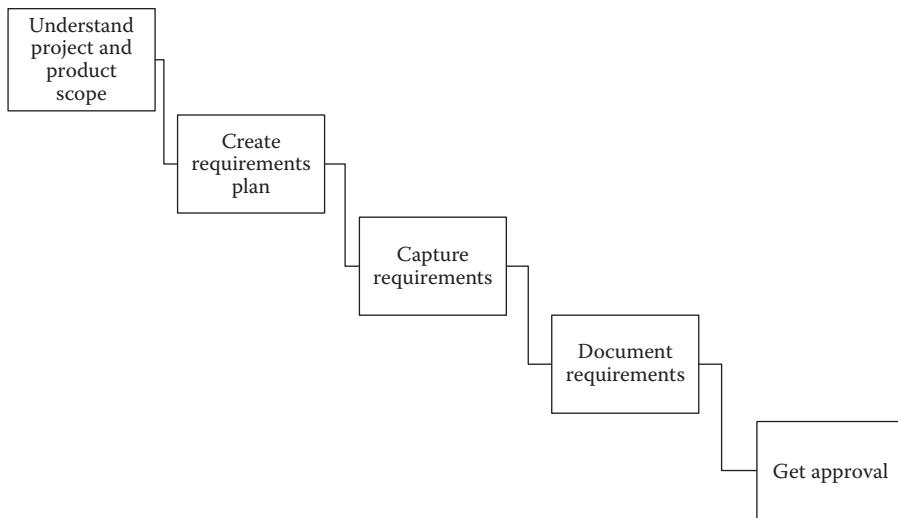


Figure 4.1 Waterfall requirements.

effort is to understand the as-is situation in the business: what the problem is and what the root causes, the organizational constraints, and the critical success factors are.

Step 2 involves creating the requirements plan (which will be covered later in this chapter). This step will let everyone know what approach to take for gathering requirements and achieving buy-in from the stakeholders. In a waterfall life cycle, this is based on the assumption that the project team has a good understanding of who the stakeholders are and what the business problem is. That allows us this clean step-by-step approach.

Step 3 captures the requirements using whatever methods were selected during the planning step. This is often the step that the business analyst starts with in a less-mature organization. The project has started and the analyst is asked to go to the customer and ask what is wanted. However, if the two previous steps have not been completed, there will be no clearly established boundaries for the requirements gathering and it will be difficult to get stakeholder buy-in to the whole process.

Step 4 is documenting and validating the requirements. Part of the IIBA definition of a requirement is that it is documented. In this step, the requirements are written down, which can be done using models or text, and then reviewed with the stakeholders to make sure that the requirements were understood correctly. In this book, this document is referred to as the business requirements document (BRD). A sample template is included in [Appendix B](#).

Step 5 is getting approval from the stakeholders. Auditability and traceability require formalizing this process. Stakeholders are often uncomfortable with actually signing off on any deliverables; they would like to retain flexibility. This tends to lessen over time if the sign-off is done consistently and if it is a part of the original plan. This step signifies the official end of the requirements-gathering phase and the start of the design phase.

When creating a task list for an analysis effort, all of these can be further decomposed into more detailed tasks such as scheduling an interview, conducting an interview, and following up on interview results. For the purpose of discussing the waterfall approach, the level of detail used above should be sufficient.

As shown in [Figure 4.1](#), it is a very distinct process, with clear tasks building on the result of the previous one. In its purest form, there is no overlap between the steps; rather, the output of each step is an input to the next. It is a manager's dream approach—clean, clear, and manageable. It is an excellent approach when dealing with well-known products and customers that clearly understand what they want. If that is not the case and the requirements gathering is more of a discovery process, then the waterfall approach can be cumbersome and stifling. Because the intent is not to revisit a decision that has already been made, a great deal of effort must be spent ensuring that the stakeholders understand and buy in to the key deliverables of each step. If that does not happen, extensive rework and poor customer satisfaction will likely be the result when the product is delivered.

When capturing the requirements, decide what level of detail is desirable. Some detailed requirements should not be captured until there is an understanding of what the design will look like. If the business analyst pushes for a decision about detailed requirements too soon, it will lead to the customer making up requirements or not thinking through what is really needed. This tends to mean that in a waterfall approach, some of the detailed requirements will not be captured until the design phase, which actually shows that even a waterfall approach will have elements of iteration in it. There is a more in-depth discussion in [Chapter 8](#) about the desired level of detail when gathering requirements.

When to use

- For large projects
- When there are well-understood product/requirements
- With proven technologies
- With enterprise solutions

Advantages

- Easy to plan and manage
- Sign-off at each step
- Sets customer expectations up front
- Limits revisiting the same issues throughout the project
- Can easily be integrated with other initiatives at a corporate level

Disadvantages

- Tends to limit customer involvement.
- Focus is on retrieving requirements rather than discovering them.
- Misunderstandings are often not realized until the end of the process.
- Can cause extensive rework.

Tips to make it successful

- Have stakeholder sign-off at the end of each major step.
- Use prototyping and modeling to ensure that each stakeholder understands the results of each step as it is completed.
- Focus on key requirements and do not go too deep too fast.
- Realize that even with this approach, there will be an element of iteration to it.

4.3.1.4 Iterative Although there have always been projects performed using iterations (sometimes even on purpose), it was Barry Boehm who formalized the iterative, and later the spiral, life cycle. For the purposes of the analysis phase, the spiral life cycle is considered to be just a specialized version of the iterative life cycle (which is a simplification, but an acceptable one). [Figure 4.2](#) gives an idea of the two approaches.

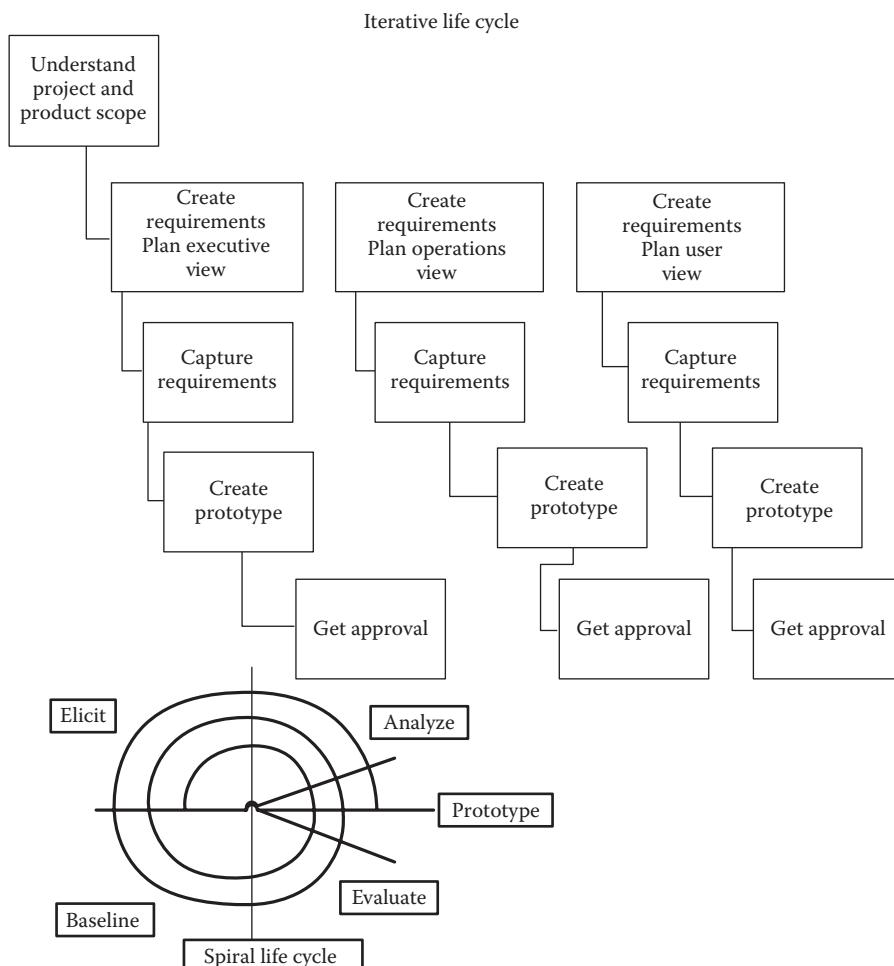


Figure 4.2 Iterative and spiral life cycle.

Instead of the nice, clean flow of the waterfall, each activity is now revisited multiple times, albeit with a different objective each time. Looking at the pictures, it is easy to see that from a project management view, this may be a difficult process to manage. However, it is often a very natural approach to a business analyst. The gradual uncovering of requirements, gradually learning more about the customer needs and the resulting requirements, tends to mirror the way most people will learn any activity. Start simple and expand.

The first step is to understand the product and the project, identifying the goals and the objectives. This will often be at a high level for an iterative approach, but it is necessary to ensure that there is an understanding of the final destination. Without overall product vision and goals, each of the iterations will eventually move in uncontrollable directions. This first step is the same start as in the waterfall model. Do not underestimate the importance of this activity; it may be the most critical one to the success of the iterative approach. There must be a framework established that shows what it is that the future iterations will accomplish. Not only should this show what the end product will be (though at a high level), it should also show the unique goals and objectives of each of the iterations.

Without overall product vision and goals, each of the iterations will eventually move in uncontrollable directions.

Next, the first iteration is initiated—in this example, the executives’ view. The iteration goes through the four-step approach of planning, elicitation, documentation, and review and approval, often accomplished in the form of a prototype. At the end of the iteration, the decisions and documents created should be baselined. It does not mean that changes are not expected, because they are, but any changes to a previous iteration should be formally documented and be approved by the stakeholders from that iteration. In the example in this book of C.V. Green and the Prescription Interaction Project, this first iteration would focus on the executives of C.V. Green, possibly including directors of marketing, sales, regional centers, and corporate quality functions.

The process is then continued into the next iteration and the next level of stakeholders, the operations’ view. Here, the store managers, pharmacy managers, and other first-level managers would get involved. The steps above are repeated, within the context that was established during the first iteration.

Then, finally, the effort goes to the user’s view. This could be the pharmacist or the clerk in the pharmacy. At this level, the key functionality should already have been established and the focus is on how the user interfaces and the detailed requirements will look. There is no theoretical limit on the number of iterations. For the example above, there could be an integration iteration, looking at integration with accounting, distribution, and purchasing. There could also be a support iteration and possibly

others added as well. However, because iterations tend to lend themselves to scope changes, it is good to minimize the numbers.

Basically, the product is gradually built and the customer verifies each level of requirements as they are being captured. This step-by-step approach ensures buy-in throughout the process, and it also lets customers gradually refine requirements based on what they learn as they go through the process. The focus of each one of the iterations can vary from project to project, but it must be predetermined. Starting an iteration without clear goals and objectives is likely to result in chaos in the end.

When to use

- For medium to large projects
- When requirements are unclear
- For research and development
- With new technologies

Advantages

- Makes it possible to learn from previous iterations.
- The customer sees the product evolve.
- The customer and business analyst will follow a logical path to create new ideas.

Disadvantages

- Each iteration may rework the previous.
- It may be difficult to contain the scope.
- It is difficult to know when to stop iterating (like the old Energizer bunny commercial, it just keeps going and going).

Tips to make it successful

- Establish change control early in the process.
- Baseline the outcome of each iteration.
- Treat each iteration as a mini-project with objectives, tasks, and deliverables.

One popular implementation of an iterative life cycle is the Rational Unified Process® (RUP®), owned by IBM. It is both a framework and an actual product. It is customizable and has a lot of helpful tools for developers. It is primarily for larger organizations and larger projects because the learning curve can be high and the cost of the product can be significant.

4.3.1.5 Agile The hottest (or coolest, depending on your vocabulary) concept in development methods today is agile development. It sounds neat, it sounds state-of-the-art, and it sounds a bit dangerous. It can be all of those things. Agile approaches are great for many development projects, but like all other approaches that have been developed in the last few decades, it is not the end-all solution to all software development problems. Keeping that basic premise in mind will make it much more likely that the agile approach will succeed.

So what is it? Agile is really a grouping of a number of approaches and techniques that appeared on the scene mostly in the 1990s, such as XP, Scrum, and the Dynamic Systems Development Method (DSDM). Rapid Applications Development (RAD), which came onto the scene in the mid-1980s, can, in many ways, be seen as a precursor to the agile approaches, and for the purposes of this book, it will be grouped with them. Developers were contrasting the long development cycles and the rigors of the waterfall and the iterative life cycles, by focusing on making software releases that were smaller and more manageable. It is often seen as a risk mitigation strategy, and rightfully so. Studies have shown that one of the largest drivers of failed projects is size. The larger the project, the more likely that it will fail. So, making the projects smaller and more manageable makes sense. Does that mean that agile is only for very small efforts? Not necessarily. Even a large project can often be broken down into smaller subcomponents, especially if the approach is to do only a portion of the project using agile methods; for example, requirements gathering may be agile and the rest of the project waterfall, or vice versa.

The larger the project, the more likely that it will fail. So, making the projects smaller and more manageable makes sense.

4.3.1.5.1 Rapid Applications Development When RAD became a popular approach in the 1980s because of books such as James Martin's *Rapid Application Development*, it was really a grouping of a set of tools and techniques that could be combined to increase project success. [Figure 4.3](#) shows the three cornerstones of RAD—process, tools, and people—with the premise that the only way to be successful is to focus on all three areas.

At that time, there was a tendency in the industry to believe that if there just was a better tool, all the development problems would go away. Development tools were getting more sophisticated, especially with the onslaught of a multitude of computer aided systems engineering (CASE) tools. Although some of the tools were very good, the danger with all of them was that they often focused too much on the tools part

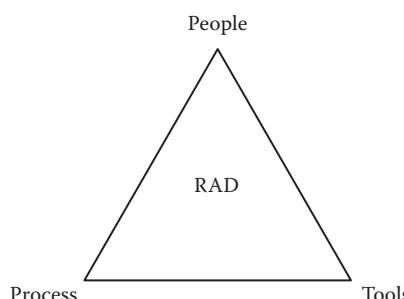


Figure 4.3 RAD cornerstones.

of the development, often forgetting about the development process and the skill sets of the people using the tools. The mantra became, “If you just use this tool, all your problems will go away!”

RAD discouraged this *silver bullet* belief and promoted a more complete, three-pronged approach.

People: It always starts with competent people. By having small, dedicated skilled with advanced tools (SWAT) teams, typically five to seven people, who work together on multiple projects, learning the processes and tools and learning how to work together, productivity can be dramatically increased.

Process: Having a well-defined process will make the competent people even more productive because they can take advantage of their lessons learned and spend less time trying to figure out how to get started on a new project. Part of the process relies on checklists and roles and responsibilities that incorporate past lessons learned.

Tools: With the right people and the right process, a good tool can now be utilized. The people will be trained on it, and the tool will support the process that is being used. Instead of the tool driving the development effort, the tool is supporting it.

Although RAD was popular in many organizations, the reason was not always good. Often management in an organization would complain that development was too slow, too costly, and the quality too low. So they called it RAD, cut the expected duration in half, and expected a miracle. The miracle rarely happened. RAD does work, but it has many components. It assumes a stable organization and process. It utilizes techniques such as time boxing (setting a fixed time frame for development or requirements gathering, and limiting the scope to activities that fit into that time frame), evolutionary prototypes, JAD sessions (discussed in [Chapter 5](#)), and CASE tools. The expectation was that if these techniques were implemented and supported, the organization would see a dramatic productivity increase. However, productivity increased not on the first project, but rather on the second or third attempt, when the organization was up to speed on the tools and techniques and had learned what pitfalls to avoid. Like so many approaches, RAD needs patience.

4.3.1.5.2 Scrum™ One of the most endearing things about agile approaches is all the cool names that are used for the techniques. Scrum (named after a rugby term) has its roots in object-oriented development, even though most of the practices are usable in any environment. It focuses on short cycles, often 30 days or less, during which a specific deliverable is created. These cycles are called sprints. From a business analysis view, the requirements are defined by the business owner in the beginning of the sprint and stored in a backlog of work to be done.

In general, Scrum assumes that requirements are typically not well understood early in the process and that they are likely to change during development. Also, the development process tends to be unpredictable, with frequent surprises. To counter this, deliverables are smaller and more manageable. Each deliverable may be a piece of software, a model, or a prototype. Either way, it is small enough to be well understood by all the people involved.

There is also a strong focus on communication between the development team and the other stakeholders. For example, there is a daily status meeting, often called a *stand-up meeting*, with a time limit of 15 minutes. These meetings are led by the ScrumMaster, the team facilitator, who must make sure that the meeting is focused and stays on topic.

There are three questions each team member must cover in each meeting:

1. What tasks have you worked on since the last meeting?
2. What are you planning to work on next?
3. Is anything blocking you?

The purpose of this meeting is not just status, but to make the work of each team member transparent to the team and to other stakeholders. It allows *bad news* to be raised quickly, identifies obstacles that other team members may be able to help with, and lets everyone know of any changes to deadlines or expected completions. Because agile development is less formal, these daily communications become critical for success.

Another key role of the ScrumMaster is to remove obstacles, shield the team from external noise, and in general provide a productive environment. Anyone exposed to a large corporate environment knows that there are many distractions for a project team. Most of them are not even related to the project, such as meetings, reorganizations, personnel changes, and announcements. The ScrumMaster's job is to minimize the team's involvement in those activities.

Most of the Scrum process is concentrated on the development rather than the requirements portion of the project; however, it is important for the business analyst to understand the Scrum approach. Even though the requirements are defined up front, the assumption is that they will change so the business analyst will need to be involved throughout the process. In addition, the analyst will be a part of the sprint reviews to make sure that the right functionality was created.

4.3.1.5.3 Dynamic Systems Development Methodology The DSDM is an extension of RAD. It involves taking RAD from the large system, using the Integrated CASE (I-CASE) tools approach, and making it fit for a more discovery-oriented approach. The concept centers on tight deadlines and budgets, and uses some key principles such as empowered team and heavy user involvement, as well as frequent delivery of products that are *good enough* to be used. This *good enough* approach works when the user is involved and can see the effort moving from *good enough* to *perfect*. This approach is also seen in the sequencing and management of project tasks. Tasks overlap, and each task is started as soon as there is enough information from the preceding task. Testing

is done throughout the process; every deliverable is tested as it is completed. [Figure 4.4](#) shows the five stages of the DSDM project cycle.

Stage 1, *Feasibility Study*, looks at the project's ability to meet the business need and also looks at whether the DSDM is the right approach for the project. Workshops are held to evaluate the risks involved with the project and with using the DSDM. In stage 2, *Business Study*, the stakeholders again meet in workshops to define the requirements of the product, and time box the development effort (set a schedule for each product release with a scope for the release). Stage 3, *Functional Model Iteration*, takes the requirements and builds a functional model or prototype that will eventually evolve into the product. The business analyst is heavily involved with these first three stages. Stage 4, *Design and Build*, is the product development, and stage 5, *Implementation*, does just that.

When the requirements (both functional and nonfunctional) are identified in stage 2, they are also prioritized. These prioritized requirements are then matched with a time slot or available schedule to create a prototype plan. The process relies on functional prototypes that are developed and tested before more detail is added. Most prototyping is usually meant to be thrown away, something that is a quick-and-dirty draft; however, in this case, the prototype is intended to eventually become the system, which was also the case with prototyping for the RAD approach. The team's reward system is based on being rewarded for product completion.

Another key component of DSDM is the risk log. Risks are continually being updated and evaluated. In any iterative approach, risk is an important consideration, but even more so when evolutionary prototyping is being used. Although a clear advantage of evolutionary prototyping is that all the work being done on it will actually be used in the final product, it is also a risk. Any issues, requirements, and conflicts that are not caught early in the process will be more expensive to find and correct later on. There is a danger

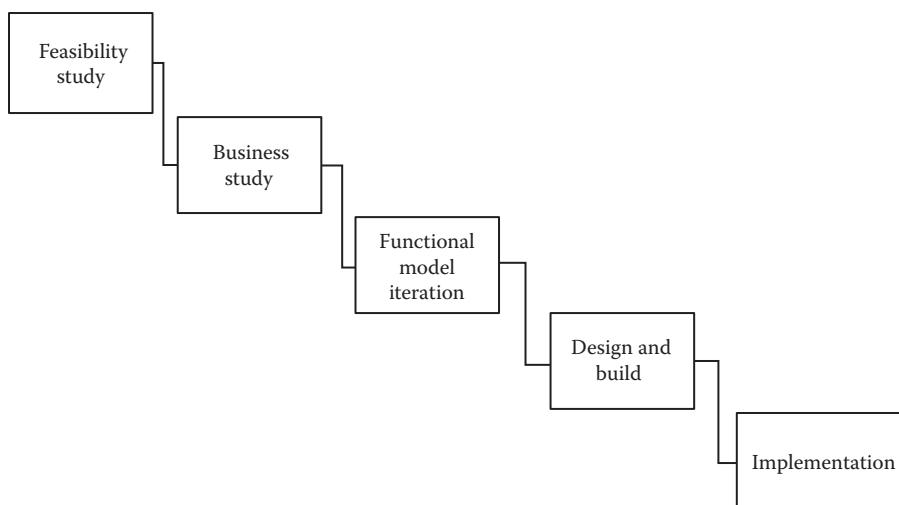


Figure 4.4 DSDM stages.

with functional prototypes when needed changes are not made because too much work must be redone. Whenever these issues come up, a risk assessment should be done to determine what happens if the changes are done versus if they are not done.

In evolutionary prototyping, any issues, requirements, and conflicts that are not caught early in the process will be more expensive to find and correct later on.

4.3.1.5.4 Just-in-Time (JIT) Requirements Gathering Scott Ambler discusses a different approach called *JIT model storming*, and it can be used with any of these other processes. This approach is still based on capturing high-level requirements early in the process, but the detailed requirements are worked out when it is time to develop a specific piece of software. With the high-level requirements in mind, a subteam involving both users and developers spends a relatively short amount of time modeling what the functionality of this portion of the system should be. This allows the team to be much more focused on a well-defined piece of the product. There is less need for large requirements documents because the developer, the business analyst, and the customer work closely together and define, develop, and deliver the product or portion of the product in a very short time frame.

Looking at the Prescription Interaction Project, this approach would probably be difficult because there would be users all over the world, they would have local differences, and there would be a need to have a more formal requirements document that could be reviewed and signed off by everyone. However, the approach would lend itself better to a subset of the project, the development of the online entry of prescription drugs, for example. There would be few subject-matter experts, the requirements would be under their control, and it would have a relatively low impact on the rest of the corporation because the business function of entering prescriptions is still the same, only the technology changes.

Some of the advantages that are seen with this approach are

- Minimizing unneeded requirements, because the focus area is small and there is less of a theoretical “what could you ever possibly want” approach and more of “what do you need right now” approach
- Better interaction with the stakeholders by limiting the scope of the discussions

However, there are also some difficulties that may be discovered:

- There are often difficulties trying to get stakeholders together for short, small meetings when the business uses common systems across many functional or geographical areas. JIT works best when stakeholders are co-located.
- Dealing with conflicting requirements coming up in different JIT sessions can be cumbersome. Typically, there are different people involved with each session so consideration must be given to undoing decisions already made.

With all that, there are many great applications of this and other agile approaches. As implied by the name, agile approaches are agile and, as such, are likely to keep evolving with new methods being added. All of them add value to the thought process of capturing and documenting requirements; none of them should become an obsession.

4.3.1.5.5 Agile Summary This is really just scratching the surface on agile approaches. Many of the techniques used can also be used very successfully with both waterfall and iterative approaches. Other techniques falling into this area are

- Prototyping: Creating quick versions of the system to reduce abstractness (more on prototypes in [Chapter 6](#))
- eXtreme Programming (XP): Focus on short cycles with two people working together on all deliverables, increasing speed and quality

In general, for all the agile approaches:

When to use

- Small projects or projects that can be subdivided into small pieces
- Projects where it is necessary to constantly adapt to what is learned during the development cycle
- Small teams
- Teams that are co-located according to the PMI

Advantages

- Flexible
- Limits documentation
- Keeps the users involved

Disadvantages

- Focuses on delivering small pieces of software; may not see the big picture
- Sometimes used as an excuse not to do planning and analysis
- Requires training

Tips to make it successful

- Frequent communication with all stakeholders
- Organizational commitment and prioritization

4.3.2 Plan Stakeholder Engagement

All work done by the business analyst in the project-planning arena is viewed as a subset to the overall project plan that is managed by the project manager. The stakeholder analysis is a good example of this. The business analyst is viewed as a sub-project manager, at least on large projects. Thus, the tools used by the business analyst here should be the same tools (and templates) that are being used for the overall project plan. In [Figure 2.2](#) in [Chapter 2](#), the responsibilities for each project phase were

defined. That definition determines who the sub-project manager is for each phase. That person should use the project stakeholder analysis as a starting point but will often need to go in on a lower level of detail to determine the people actually involved with the requirements process.

There are many drivers of the requirements plan; however, the most important one tends to be the stakeholders that are involved. If this is a single-user environment with a well-defined business process, the plan should be very simple and straightforward. However, if the customer has locations on multiple continents, is operating under a new business model, and is not sure what they are really looking for, then a more detailed plan is needed.

Start the stakeholder analysis by documenting the categories of stakeholders. These categories can be overlapping (one stakeholder can be a part of multiple categories), such as a sponsor who may also be a user of the product, or they can be hierarchical (one stakeholder can be a subset of a different stakeholder category), such as data entry clerk being a category of user. Regardless, the stakeholder identification must be as complete as possible. A stakeholder overlooked at this early point of the project can lead to a significant rework or even failure of the project later.

[Figure 4.5](#) shows an example of stakeholder categorization. This stakeholder diagram should be reviewed with all the stakeholders identified, if possible. The government, as an example, may be a stakeholder, but reviewing anything with the government may not be realistic. In the review, the analyst looks for buy-in that the stakeholders are agreeing to participate in the project and also for identification of potential stakeholders that may have been missed.

Once a list of stakeholders has been documented and reviewed, the stakeholders should be prioritized. Each stakeholder's role and priority must be documented. Prioritizing the stakeholders may sound like a political activity, and it definitely can

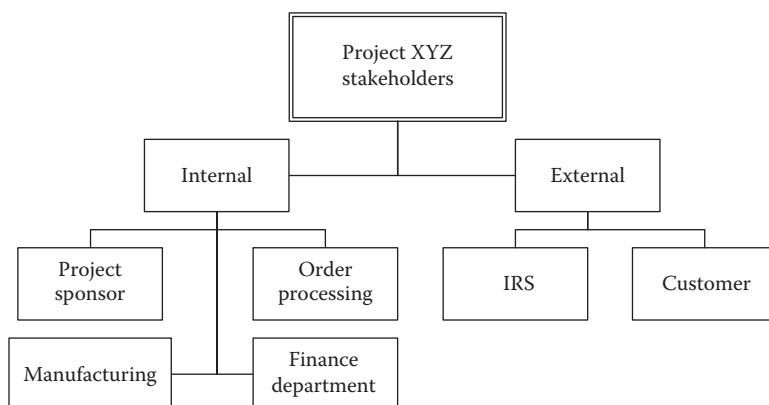


Figure 4.5 Stakeholder hierarchical view.

be, but it is also necessary. Not every stakeholder has the same influence over project decisions, and the business analyst must know, going into requirements sessions, who the key players are. The priority decision should be made by the project sponsor or the paying customer.

Why is this important? An example from earlier in my career was the development of computer systems for Saturn automotive dealers. Some of the key stakeholders were Saturn Corporation and the actual Saturn dealers. They would often have different thoughts on what the product should (and should not) do. There was a lot of discussion on control, privacy of business information, and what can be required of an independent business. Those were questions far beyond the job description of a business analyst, and they had to be decided on by the customer up front.

In this specific case, it was eventually determined that the priorities were different, depending on the part of the system we were talking about. For any functionality that was dealing with cross-dealer communication and communication with corporate, Saturn corporate was the lead. Any functionality dealing with operations of the dealership was decided on by the dealer group. If this had not been realized and agreed to, it is likely that the product developed would not have satisfied either party.

Also make sure that the stakeholder's role is well understood. Many of them may be passive; they need information from the product or they provide information to the product, but they have no real interest in what the product does. Others may be severely impacted in how they do business and may experience cascading effects, causing new projects in their area as a response. Their role also often depends on the person or persons who are selected to represent them. A strong, competent representative tends to ensure that the organization's interests are looked out for; a weak, uninterested representative can be a disaster for the stakeholder and potentially for the project. Always try to influence which stakeholder representatives are selected for the project by defining what their roles will be and what type of person is expected. Be aware of the person available. When the stakeholder states, "You can have Joe over there in the corner as our representative; he doesn't really have anything else going on," it is time to start negotiating. There is an old saying: "If you want something done, look for a busy person," and that is equally true for requirements gathering. Find a representative who speaks for the organization and is well respected.

Figure 4.6 shows an example of a completed stakeholder analysis. As seen when reviewing the form, there must be some judgment used regarding what to put in this document. Decide how it will be used. If this is private documentation, there can be more items included on the form. If this is a project form, visible to everyone, be more restrictive. In either case, naturally, do not put any inappropriate information in the form.

Project name: XYZ Project	Date: Jan 1, 2007
Project manager: John Jones	Project sponsor: Jans Smith
Stakeholder organization:	
Finance	
Stakeholder name and contact information:	
Linda Williams, VP Finance (121) 555-1212	
What will this stakeholder provide to the project?	
Will provide initial funding by month Will approve/reject change requests that require increased budgets	
What will the project provide to this stakeholder?	
Timeline for expense Cost/Benefit analysis	
What is the impact to this stakeholder if the project succeeds or fails?	
If this project fails, Linda will be responsible for finding other areas of the organization that can make up for the shortfall	
Hot issues for this stakeholder	
Solid estimates No padding at project level. She will set a management reserve	

Figure 4.6 Stakeholder analysis.

4.3.2.1 User Profiling Before any realistic plans for the analysis phase can be developed, there must be an evaluation of the scope of the elicitation effort. Some projects may only need to interview one customer; others may need to build consensus among thousands of users over multiple continents, speaking different languages and having cultural differences. Not only will these differences drive the timeline and the costing of the analysis activities, but they will also determine the skills needed by the business analyst and the necessary organizational involvement.

Users can be categorized in many ways, including

- As primary or secondary users
- As experienced or novice users
- As full-time or part-time users
- As organizational entities
- As internal or external users
- By language and cultural differences
- By geographical differences

There are many other ways as well. One good approach is to sit down in the beginning of the project and have a brainstorming session to determine who will use the system, and then review the result of that session as a starting point for the user categories.

The result of user profiling will drive not only the planning of the analysis phase, but also the actual elicitation, documentation, modeling, and communication of the requirements. It may even necessitate different solutions for different users. The information covered on the next few pages will serve as a basis for many of the following chapters. The different categories will be reviewed in the context of the Prescription Interaction Project, described in [Chapter 1](#).

The result of user profiling will drive not only the planning of the analysis phase, but also the actual elicitation, documentation, modeling, and communication of the requirements.

4.3.2.1.1 Primary or Secondary Users If the users' jobs are prescription entry clerks, then they will be primary users when a prescription entry system is developed. They cannot do their jobs without the system, or at least it is an integral part of the core job functions. This group normally comes to mind first when identifying users because they will usually be the ones spending the most time using the system. For the Prescription Interaction Project, this is primarily the pharmacist and pharmacy assistant. However, make sure to expand the thinking when looking for primary users. If the intention is for the customers to be able to enter prescription information online themselves, then that would make the customers primary users as well.

So who would be the secondary user? It could be the actual customer who wants to get a prescription filled. Or it could be the manager who needs to measure prescription entry performance or the doctor writing the prescription. The term *secondary user* does not indicate less importance or less interest in the outcome. Obviously, the customer presenting the prescription may have more interest in a well-executed order than the person entering the prescription. Secondary just means that the system itself is not a primary part of the person's job. To identify a list of potential secondary users, ask the following questions:

- Who will be impacted by the system?
- Who will impact the system?

4.3.2.2 Experienced or Novice Users Experienced users tend to be good at giving big-picture views. They understand the process, and they often understand how their role fits into the overall business. But sometimes they struggle with the detail requirements. Because they are experienced, they tend to make a lot of assumptions about the business analyst's understanding and the basic functionality of the product. The pharmacist who has been working for 15 years filling prescriptions is doing many of the mundane steps, such as verifying past prescriptions, without thinking about it, and may forget to tell the analyst about the steps in detail.

Novice users know what their struggles are, such as what steps in the process are difficult or where the bottlenecks are. They struggle with things for which the

experienced user has already discovered workarounds. At the same time, they have little knowledge of the big picture, such as why something is being done or who is dependent on the process. The newly hired pharmacy technicians will typically have a written set of instructions that they follow, even if they do not fully understand why they need to do it. Their ability to visualize a new solution is probably also more limited than the experienced user's.

So when planning for the analysis phase, there must be a plan for including both experienced and novice users, and each probably should be handled using different elicitation techniques: job shadowing for some, facilitated sessions for others, and surveys for still other users. There is a more detailed discussion on each technique as well as their pros and cons in [Chapter 5](#).

4.3.2.3 Full-Time or Part-Time Users This is not referring to the work status of the users. Instead, it is looking for how much of their time they spend using the product. A user who spends most of the day using a system will be focused on efficiency and speed; however, a user who only spends a few hours each month will tend to look for intuitiveness, forgiveness, and general ease of use. Some pharmacies may have a dedicated person doing nothing but entering prescriptions. At smaller pharmacies, there may be only one person doing all job functions, so entering prescriptions may be only a relatively small portion of that person's day. If the business analyst captures requirements from one but not the other, the system will be poorly received when installed.

4.3.2.4 Organizational Entities If the project is to develop a prescription interaction system, it would be obvious that the pharmacy department must be interviewed. Accounting, purchasing, legal, marketing, and others must be interviewed as well. The other entities may not have the same interests, but they are likely to be impacted or have an impact on the project. If the current marketing campaigns state that the organization allows for seamless transfer of prescriptions, it is likely to have an impact on the requirements for this project.

4.3.2.5 Internal or External Users Some systems are used by people from both within the organization and outside. If the prescription is entered in the pharmacy by the pharmacy staff, they will have requirements. If the prescription is entered online by a doctor or a customer, their expectations may be different. Also, the external user's environment will be much harder to control as far as hardware, operating system, etc., are concerned, which must be considered by the business analyst.

4.3.2.6 Language and Cultural Differences This can be looked at from two different viewpoints: (1) differences in the people who will eventually be using the product, and (2) differences in the people giving the requirements. Both scenarios will have an impact on the approach used to gather requirements. For language differences among the end users, the difference may lead to multilingual support, help functions, and

different auditing functions. It can also lead to a more graphical user interface with a stronger focus on pictures than on words.

For the people giving the requirements, it may lead to different approaches in eliciting, reviewing, and validating requirements. Brainstorming sessions are often difficult to do when the participants are speaking different languages, so one-on-one interviews or surveys in the users' primary language may be an option. For the pharmacy, this can be a difficult issue. At the headquarters location, where most of the requirements are likely to be gathered, there will often be a primary language used; however, in more rural pharmacies, the language could be different.

4.3.2.7 Geographical Differences Some geographical differences may overlap with the language and cultural issues discussed above, but they can also be related to legislative differences between different countries or even within a country, as well as issues with network, power structure (electrical power), and availability of technology. There also tends to be very different price sensitivities for product pricing in different parts of the world: what is considered affordable in Sweden may not be viewed the same way in India.

4.3.2.8 Stakeholder Communication The communications plan is partly an extension of the stakeholder analysis. It focuses on status reporting to the project manager, customer, management, and others, as well as on information needed to create the deliverables. It can be arranged in columns covering the following:

- What—What is the content of the communication? Is it a status report? Is it approval? Is it a document? Create and show templates for the communication when possible.
- Who creates—Who is responsible for creating and distributing the communication?
- Who receives—Who will be the recipient? Verify that it is really wanted or needed.
- Why—Is this really needed? What is being accomplished with this communication? The reason for the communication should be understood by all.
- When—When will this happen? Once? Weekly? At the beginning of a phase? When a milestone is reached?
- Where—The location for meetings.
- How—in person, videoconferencing, over the web.

Figure 4.7 shows a partially filled out communications plan. The communications plan serves two main purposes for the analysis phase. First, it is a to-do list and, like any plan, it is a trigger for what to do and when. Second, it lets everyone else on the project know what is expected and assists in getting them to buy in to communication. If the communications plan states that the customer will approve the requirements document within 15 days, and the customer signs off on the communications plan, the chances of actually getting customer approval increases dramatically. Project

What	Who (responsible)	Who (audience)	Why	When	Where	How
BRD approval	Business analyst	Customer, developer, sponsor	Go/No-go decision for the analysis phase	April 10, 2007	Executive conference room	In person

Figure 4.7 Communications plan.

management and business analysis are both communicative responsibilities. The communications plan lays the foundation for that communication.

4.3.3 Plan Business Analysis Governance

The governance of the business analysis activities are at least partially overlapping the governance of the project management activities. When governance is created from a business analysis view, change control is often the natural focus. Who can approve product changes, both from a functionality view but also from a cost and scheduling perspective?

It is, however, also to look at it from a risk view. Have enough stakeholders been interviewed? Have we formed a consensus? Do we have a good enough understanding of requirements before we move into the next phase? While this process is owned by the project manager, the business analyst will be a critical resource for analysis and recommendations.

The end result of the analysis phase is the BRD. This is the ultimate definition of the product scope. As such, it must be controlled throughout the project and any changes to it should be evaluated to assess their impact on other project and product objectives.

Even though the discussion here talks about the BRD as one document, it is really a composite of many documents. Change control is initiated at the time a document is being baselined. The different sections of the BRD should normally be baselined at different times. For example, the executives' requirements should be captured and baselined before the user requirements are finalized, and the user requirements should be baselined before the system requirements. So part of the change control process should be to establish on what documents the project will track change control and when those documents go from being *work in progress* to *baseline*. Many organizations baseline the whole BRD at one time, and although this sometimes works fine, it is not consistent with a progressive elaboration approach. The approach to baseline or *freeze* different parts of the requirements document at different times does add complexity to the change control process; however, on a complex project, it minimizes the chance of scope creep as you go from one stakeholder type to the next.

In line with this approach, there must be a decision up front with regard to who can approve what type of change. A change that has a cost impact may only need to be approved by the customer; however, a change that will delay the schedule may also need approval from the project office (to evaluate impact on other projects that may need the same resources). Establish responsibilities up front and follow a repeatable process that is known to all. It would typically look similar to this:

1. Submit change request.
2. Assign for analysis.
3. Evaluate impact of change.
4. Forward request with recommendation to change control board.
5. Approve or reject.
6. Track changes.

Successful change control is largely about discipline. All changes must go through formal change control. This does not mean that the customer will be charged for all changes. Billing is a business decision; impact analysis is an awareness issue. Successful change control will ensure that changes that are made are needed, wanted, and meet the objectives of the project. Despite customers feeling that they are being hindered by all this paperwork, it is really there to protect them from changes that are not desirable to the organization (even though they may be wanted by a stakeholder).

4.3.4 Plan Business Analysis Information Management

The various documents created during the business analysis process need to be carefully designed and reviewed. What level of detail is needed? What information is needed by whom? The two main documents (they will be expanded on later in the book) are the business analysis plan and the BRD. They may have different names in some organizations but they do define

- What process will be used to effectively elicit, structure, verify, and approve requirements
- What is the information needed by the various stakeholders to understand and make decisions about those requirements

4.3.5 Plan Business Analysis Performance Improvements

Metrics and process evaluation are part of any project. In my experience, most metrics are taken during the later phases of the project life cycle. Therefore, it is nice to see that the IIBA is putting special emphasis on this. This task gives attention to establishing what metrics are needed to evaluate the quality and effectiveness of the business analysis process. There are three main elements of this:

1. Establish the measurements needed.
2. Report on the performance.
3. Decide what to do about the performance.

The benefits of this task are, first, that it can improve the performance for the specific effort that is being undertaken and, second, that it can be used for ongoing process improvement.

4.4 Business Analysis Plan

On a large, complex project, there needs to be a comprehensive plan for the analysis phase; for a small project, most of the plan may be Post-It® notes on workstations. Project planning is not a *one size fits all* activity. Using a repeatable process and pre-defined templates will make the planning process less painful; however, any process selected must still be customized to fit the specifics of each situation.

Using a repeatable process and predefined templates will make the planning process less painful, but any process selected must still be customized to fit the specifics of each situation.

The following shows a sample requirements plan template (see [Appendix B](#)) with the key sections and a brief description of each. After that, a more detailed review of each section follows:

1. Project overview and background. This provides context and background for readers.
2. Scope and deliverables. This shows scope and deliverables from the viewpoint of a business analyst.
3. Stakeholder analysis. Who is impacted by (or is impacting) the project?
4. Communications plan. How will project status be identified and communicated? What types of meetings are needed?
5. Project activities. What work needs to be done on the project? This could be in the form of a work breakdown structure (WBS) or in a list format, and will be input into detail scheduling and estimating.
6. Roles and responsibilities. Who will do what on the project?
7. Resource plan. What resources are needed for the completion of the analysis activities? This includes human, software, and equipment resources and tools.
8. Requirements risk plan. What are the requirements/product-related risks?
9. Manage changes to requirements. How are changes handled?

Item 3, stakeholder analysis, is covered above, in [Section 4.3.2](#). The communications plan, item 4, is reviewed in [Section 4.3.2.8](#). Some organizations may require additional sections, such as a quality plan or a procurement plan. Negotiate with the project manager to determine what is needed for the project.

4.4.1 Project Overview and Background

This is not intended to be a duplication of the project information that the project manager has collected. The general project information may be referenced here, but should not be copied. Duplication of information tends to cause confusion and should be discouraged. Rather, this section should focus on how the developed product fits into the overall enterprise. How was the original need identified? How will it fit into the product portfolio? What is the long-term vision of the product? A lot of the information covered here would be developed or captured during the enterprise analysis described in [Chapter 6](#). Think of including the following:

- Origination of the project idea
- Previous attempts to develop this product
- Competitive forces
- What success will do for the organization
- What failure will do for the organization
- Initiatives that this project is dependent on
- Other initiatives that are dependent on this project
- Key business integration point

This section may have many uses, but one of the main purposes is *lessons learned*. When future projects try to learn from this effort and evaluate what worked and what did not, this section will help put things into perspective. It is also a good training tool for new people joining the project. It gives them an idea of the history behind where the project is today.

4.4.2 Scope and Deliverables

The project scope will be documented in the charter, statement of work, WBS, or other documents, depending on the process used. This section is an extension of previously completed documentation into areas that are of significant interest to the business analyst. Typical deliverables are a BRD, a requirements plan, a user acceptance plan, and others, depending on the specifics of the project. This is intended for establishing boundaries, what is included and excluded from the scope of the project, again with a focus on the analyst's viewpoint.

Examples of good areas to explore for inclusion or exclusion are

- Geographic locations—Is there a need to talk to stakeholders from multiple locations or countries? Will the users all speak the same language?
- Staged deliverables—Will these be requirements for the final product, or just for a portion of that product?
- Integration to other areas of the business—Whose responsibility is this? Do the people in each area need to define the impact on their systems?

Also look for responsibilities of the business analyst that may go beyond the analysis phase. Who will train the user? What is the business analyst's involvement with testing? What kind of follow-up is needed after implementation? These are all activities that need to be planned by their owner. It actually may mean that the business analyst's scope goes beyond the project scope. The project scope may end when the system is implemented, but the business analyst often has tasks going past implementation. Examples include assessing customer satisfaction, training users, and quantifying improvements.

4.4.3 Project Activities

The process of identifying the activities needed for the analysis phase lays the foundation for estimation, scheduling, sequencing, and overall planning. But more than that, it greatly increases the business analysis team's understanding of what the phase is actually all about. In a number of the classes I teach, the students are asked to create a WBS or a task list. Inevitably, it becomes somewhat frustrating because they do not have enough information and they are not subject-matter experts in the business areas used in the classroom. Even so, it forces them to think about the work to be done, to make assumptions, and to ask questions. It is relatively easy to understand a one-line scope statement, but it is often very hard to take that statement and create a detailed activity list. [Figure 4.8](#) shows an example of a WBS for the analysis phase of a project.

When creating the WBS, involve as many of the team members in the process as possible. Utilizing the diversity and experience of the whole team will create a more complete and accurate WBS. If the organization follows a standard process for the analysis phase, use that as a starting point. If there is not a standard process, try to find a similar project that the organization has done in the past, and use that as a starting point.

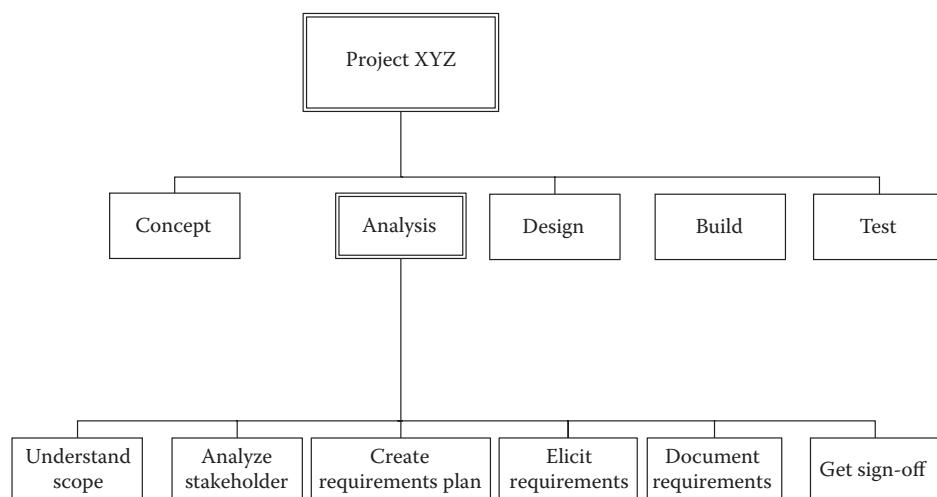


Figure 4.8 Sample WBS for the analysis phase.

Most plans should be developed using progressive elaboration as a guideline. This technique recognizes that in the beginning of a project (or a phase of the project), there is limited knowledge and many unknowns that will be difficult to plan for. By using progressive elaboration, the areas that are well understood are planned at a detailed level, while areas that need more work can be planned at a higher level. This is definitely a *best practice*; however, in some organizations, where the plan must be 100 percent complete and shared with the customer at a detailed level, this approach may not be allowed. Unfortunately, the estimating process in those organizations tends to result in more contingency (and padding) of estimates.

By using progressive elaboration, the areas that are well understood are planned at a detailed level, while areas that need more work can be planned at a higher level.

4.4.4 Roles and Responsibilities

If there is only one business analyst on the project, defining roles and responsibilities may be an easy task; however, on large projects, this becomes more complex. Responsibility matrixes (such as the one shown in [Figure 4.9](#)) can be done at multiple levels.

Early in the process, the tasks will be relatively high level and can often be assigned to an organizational unit or to a job classification (such as the business analyst or the facilitator). As the project moves along the path of progressive elaboration, the plan will go into more detailed tasks and eventually should get an actual name assigned to each task.

A responsibility matrix must be approved by the organizations and people who have tasks to perform. *Approve* may be a strong word, but they must at least acknowledge that they are aware there is a task for them on the project. Remember that many of the tasks done in the analysis phase may be done by the customer. Those tasks must also be documented in the roles and responsibility matrix, and customers must buy into their involvement. There is not much point in the business analyst creating a survey if the customer is not committed to filling it out.

<i>Task/responsible</i>	<i>Project manager</i>	<i>Business analyst</i>	<i>Project sponsor</i>	<i>Technical writer</i>
Analyze stakeholder	A	R	I	
Create requirements plan	A	R	I	
Elicit requirements		R	A	I
Document requirements		C	A	R
Get sign-off	R	I	A	I
<i>A = approve, C = concur, I = inform, R = responsible</i>				

Figure 4.9 Roles and responsibility matrix.

4.4.5 Resource Plan

Creating the schedule, figuring out skills needed, and assigning responsibilities is good progress. The resource plan shows how this will actually happen. Identifying the people needed, when they are needed, and for how long, is not an easy task. It requires communication and negotiation skills. Also, the task of identifying nonhuman resources, such as the tools, the software and hardware, and the facilities needed, is often a time-consuming and contentious activity.

Most projects today operate in some type of matrix organization. That means that although the business analyst will have responsibility for the analysis phase and will be held accountable for tasks in that phase, the resources will typically be reporting to someone else. Being in a matrix organization probably also means that the resources for the analysis work will not be assigned to the project on a full-time basis.

Start early with the resource plan and keep communicating with the organizations that will be providing resources. Common resources needed for the analysis phase are data architects, quality assurance specialists, testing groups, business process owners, facilitators, and support staff. By involving the management early on and keeping them up-to-date on project progress, the project is more likely to get the right resources at the right time. Be aware of what the project's priority is within the organization and periodically assess if that priority is changing. Although it is fairly easy to acquire resources for a high-priority project, if that priority decreases, the availability of resources tends to decrease as well.

4.4.6 Requirements Risk Plan

The PMI's *PMBOK* defines risk as "an uncertain event that, if it occurs, has a positive or negative impact on at least one project objective." Although that is certainly a good definition, for this book, the concentration is on the risks that have a negative impact. The positive risks or opportunities are of more interest to the business owner, even though the business analyst definitely has a role in identifying them. It is also fair to say that opportunities are looked at more during enterprise analysis, which is normally where the full benefit of an opportunity can be realized.

Software development is risky business. For the project manager, there are risks that deal with new and unproven technology, changing technology, lack of knowledge about the capabilities of the organization, and lack of trained people. The risk plan covered in this section focuses on the risks that the business analyst is most concerned with.

For the business analyst, there are two main areas for risk management: (1) risks associated with the actual requirements-gathering process and (2) risks associated with the product being developed. Risks associated with gathering of requirements are more temporary in nature and are primarily dealt with by the analyst during the analysis phase. Risks associated with the product are addressed throughout the

project life cycle, and these risks are really owned by the business owner, although the business analyst should be heavily involved as well. On the basis of the different audiences for these risks, it would normally be good to track and report on them separately.

Typical requirements-gathering risks to consider are as follows:

- Customers not knowing what they want. This is more often a question of not being able to clearly express what customers want, often based on not speaking the *techie* language.
- Business analysts not understanding the business. Traditionally, most business analysts come from a technical background; however, this is changing. An increasing number of business analysts are now from the business side, which should help reduce this risk.
- Lack of skills. There are more tools available for requirements gathering than ever before. This is good once the business analyst is up to speed on the tools; however, until that happens, the tools can actually slow down the process.
- Customers not willing to dedicate enough time for the requirements process. Most customers actually have a regular job to do. It is often difficult for them to find time to meet with the business analyst.
- Customers defining the solution, not the requirements. Customers often see something similar to what they want. Instead of describing what they need, they describe what they perceive as the best solution. The business analyst needs both skills and tact to push back on this.

Typical product-related risks are as follows:

- Product not meeting the business need. It may meet the requirements, but the requirements do not solve the problem.
- Users not accepting the product. This is especially common when only a subset of the user population has been involved with giving requirements.
- Product pricing not competitive. The product is great, but no one can afford it.
- Product too complex for the user. It may meet the functional requirements, but is too complex.
- Technology outdated by the time the product is released. The state of the art changes fast. If leading edge is needed, the project needs to be short.

A general risk approach, which can be applied during enterprise analysis, requirements gathering, or any other analysis activity, is discussed later in this chapter.

4.4.7 Manage Changes to Requirements

Part of any plan is to document how changes are handled. This is explained in [Section 4.3.3](#).

4.5 General Guidelines for the Requirements Plan

The next few sections will delve into more detail on specific tools and techniques for creating different portions of the plan described above. The business analyst must work with the project manager to clearly define which of these tasks should be done by the analyst, and which by the project manager. The tasks here are project management tasks that may (and often should) be delegated.

4.5.1 Risk

The risk process recommended for the analysis phase is very similar to the overall risk management process as shown in the *PMBOK*. It has been simplified here to account for it being a subset of the overall risk process. The main steps in the process are

1. Develop the risk management approach.
2. Identify risks.
3. Assess risks.
4. Respond to risks.
5. Monitor and control risks.

4.5.1.1 Step 1: Develop the Risk Management Approach How much time should be spent discovering and dealing with risks? The easy answer is “it depends.” But on what does it depend? A large project tends to be more risky than a small one. A product that may jeopardize the company’s future will also deserve a higher focus. Some people are gamblers, some are not. So the following areas are worthy of research when determining the risk approach:

- Project size and complexity
- Precedence—has anything like this been done before?
- Skills with tools and technology
- Maturity of the business
- Impact if the project succeeds or fails

On a low-risk project, the project team may spend less than 10 percent of the project time dealing with risk and contingencies; on a high-risk project, that can easily get up to 25 percent or more.

Develop a plan up front, define roles and responsibilities for the risk activities, and get buy-in from the involved parties. This minimizes the chance of over- or underfocusing on risk. Keep in mind that finding and reducing or eliminating risks is probably the most effective approach to delivering the right product in the right time frame for the right price.

4.5.1.2 Step 2: Identify Risks Identifying risks can be a time consuming and sometimes unproductive activity. Although brainstorming is the most common approach

to risk identification, brainstorming that is not focused often leads to a list of obvious risks or risks that are difficult to address. The more structure and focus there is in the process, without stifling creativity too much, the more useful the activity will be. Some *best practices* when identifying risks include the following:

- Learn from the past. History has a tendency to repeat itself; the areas that have caused problems in the past can be a good starting point. Documentation of lessons learned can be difficult to find; some project managers may have them hidden away, or in some cases they may have been thrown away. Start a new best practice within your organization. In a common area, set aside a folder called *Lessons Learned* and start filing the reports created for lessons learned in there. In a few years, this practice will yield an impressive starting point.
- Talk to the experts. Anyone who has done a similar activity in the past or who has seen similar efforts in other organizations can identify potential areas of concern. Experts may be past project managers, project participants, or external resources. Vendors and customers may have undertaken similar efforts in the past.
- Categorize risk. Categorization can be done before or after the risks have been identified; either way, it helps determine priorities. Categorization can help identify further risks, as well as identify areas that may have been overlooked.

Figure 4.10 shows a risk categorization structure.

A more detailed risk structure, covering the whole project, not just requirements, can be found on the SEI website (www.sei.cmu.edu; search for Risk Taxonomy TR06.93). This has been developed to focus on the risks associated with software development and based on industry best practices. The specific classification system used is less important than the fact that one is being used. Classification of risk helps with readability, organization, and assigning responsibilities, as well as with prioritization of risks.

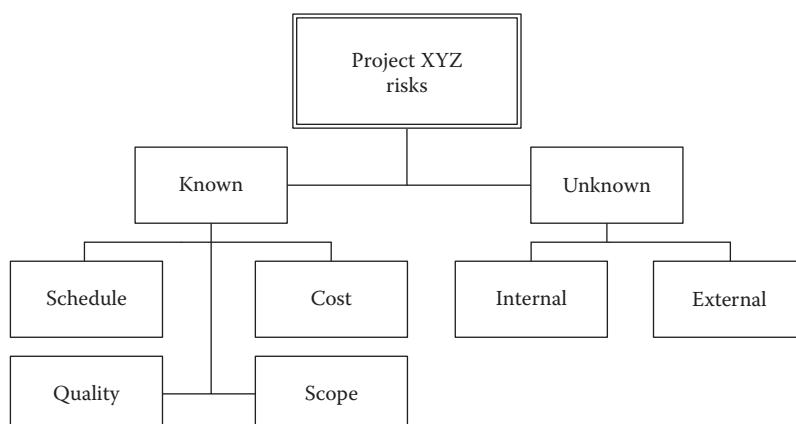


Figure 4.10 Risk structure.

Document the risks in a cause-and-effect statement. “If the customer does not attend the prototype review, the final product may not be accepted by the users.” Too often, risk statements are too generic to actually act on. Risk statements such as “May miss schedule,” “May go over budget,” and “System may not work” are not really describing what the risk is, but rather what the impact of the risk will be. The cause-and-effect approach gives a much better understanding of the risk and the main impact areas.

4.5.1.3 Step 3: Assess Risks Although it is best practice to identify a large number of risks, it is also important to spend most of the available time and money focusing on the high-value ones, meaning that the focus should be on the risks that are most likely to hurt the project objectives. The two dimensions traditionally used to evaluate risks are probability and impact: how likely is it that the risk will occur, and if it happens, what is the impact of the risk?

In this book, there is a third dimension added, called controllability. This is a somewhat subjective analysis of the level of control there is over the risk drivers. For example, a risk of “Inadequate skills on the project team will lead to poor quality” is a risk that appears to be controllable. However, the risk “FDA may pass new reporting requirements, causing our project to lose resources to other projects” is a risk over which the project team probably has limited control. It does not mean that those risks are ignored, but the thought process to decide on responses will be different.

Figure 4.11 shows risks positioned on a graph, giving an easily understandable view of the priorities. Ideally, there is enough information to put an actual percentage and dollar amount on the probability and impact, but this is not realistic for most software projects. Good history is needed to create good numbers, and for most organizations that history is just not available yet. So for most projects, it is often sufficient to use a high/medium/low approach to qualifying the risk, with some subjective definition of what high versus medium and low means. The controllability dimension will always be subjective, so high/medium/low works well for that. Using the example risk from

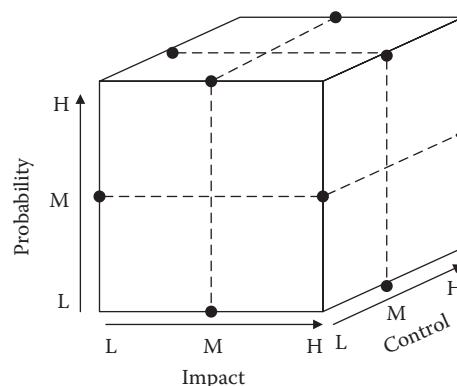


Figure 4.11 Risk prioritization tool.

earlier, “If the customer does not attend the prototype review, the final product may not be accepted by the users,” to classify probability, the analyst may find that because in the past the customer has attended most sessions when sufficient notice has been given, the probability is rated as medium. The impact would be very severe to multiple project objectives so the analyst rates the impact as high. Controllability is medium, the customer is not directly under the analyst’s control, but there are ways to influence the customer.

4.5.1.4 Step 4: Respond to Risks There are four classic ways of responding to risk:

1. Avoid: Do not do the risky behavior. If the risk is associated with prototypes, do not use prototypes (this may create other risks, but this specific risk is avoided). Just not doing the project is another example of *avoid*.
2. Mitigate: Take some action to minimize the risk. For the risk that the customer will not attend the review, it could be “Have review in user’s office,” or “Send out reminder notice.”
3. Transfer: Find someone to absorb all or part of the risk. This could involve outsourcing, insurance, and penalties in contracts.
4. Accept: Maybe there is nothing that can be done, or the price of action is higher than the price of doing nothing.

Responding to risks is often a mix of the approaches above. Depending on the project and on the specific circumstances, there may be some mitigation of a risk; some of it can be transferred while the rest is accepted. It is important to consider all options. There is a tendency to try to mitigate all risks, but this is not necessarily the right approach. If the cost–benefit ratio of inaction is better than that of action, *accept* would be the better risk approach.

Also be aware that in most cases, the response to one risk may be a source of new risks. Part of the response evaluation and documentation should be a review of what new risks have been created. This is especially true for the business analyst. The risk of the customer not understanding the requirements is mitigated by prototyping. This creates the risks of prototyping setting unrealistic expectations with the customer, which is mitigated by creating a prototype that will have no real functionality and on it goes.

4.5.1.5 Step 5: Monitor and Control Risks Monitoring and controlling risk should be done throughout the analysis phase. Normally, it can be included as a part of the regular project status reviews being done throughout the phase; however, the analysis team may need to add additional checkpoints focusing on requirements gathering and product development. It is especially important to have a checkpoint at the end of the analysis phase, document what risks are still outstanding, and make sure that the development team and the project manager are aware of them.

4.5.2 Estimating

One of the main purposes of creating a requirements plan is to help in negotiation with other stakeholders. To this end, the more reliable and justified the estimate is, the stronger the analyst's negotiating position will be. Often in a negotiation, the person with the most facts wins. A *gut feel* estimate of "I'd guess about three months for that" will not be very credible with the stakeholders. However, a step-by-step plan outlining the analysis activities with assumptions about each of them and an estimate to go along will be impressive. It tends to shift the focus from "How do you know how long that task will take?" to "Which ones of these activities do we really need to do?"—which is a great starting position.

The more reliable and justified the estimate is, the stronger the analyst's negotiating position will be.

There are two basic types of estimates you can create:

1. Analogous estimates, often called top-down estimates. These estimates are often done early on and are based on similar activities done in the past. If the last project the organization did was similar to the current one, that project result may be used as an estimate for the current effort. If facilitated sessions are used on this project, compare it to a project where similar techniques were used. It is an estimating technique that has a high degree of inaccuracy; its main use should be to see if the estimate is in the ballpark. This type of estimate may be ±50 percent, but it gives an idea with regard to its being a \$1-million or a \$10-million project. Good solid assumptions are needed for good analogous estimates.
2. Detailed or bottom-up estimates. This requires a breakdown of the analysis phase to detail activities. This is often unrealistic in the beginning of the analysis phase, because the stakeholders are unknown and the requirements have not yet been captured. It may work for a small, well-defined project; however, in most cases, there needs to be quite a bit of work done before this level of detail can be reached.

The business analyst should review the estimate with the project manager, and it would normally be the project manager who actually presents the estimate to the customer or sponsor.

4.5.3 Laying Out Tasks

Part of any good plan is to determine which activities are dependent on other activities and which ones can go in parallel. The following five activities will serve as an example:

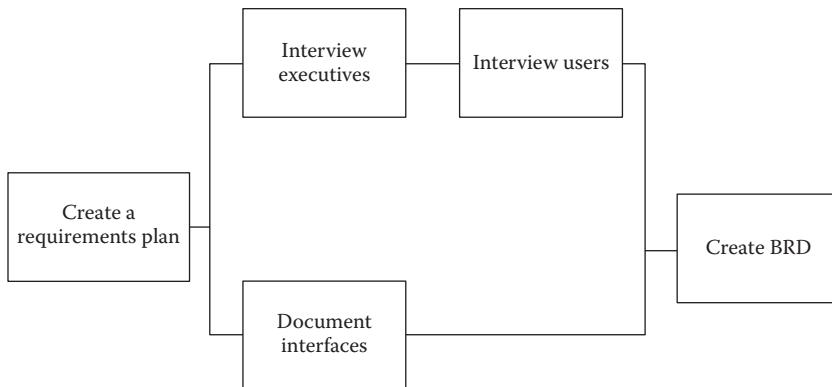


Figure 4.12 Network diagram.

1. Create a requirements plan.
2. Interview executives.
3. Interview users.
4. Document interfaces.
5. Create a BRD.

The user interviews should probably not start until the executives have been interviewed; however, documentation of interfaces can be done in parallel with both of those activities. So, this could look like the network diagram shown in [Figure 4.12](#).

The network diagram is a useful tool to analyze risks, find areas that can speed up the project, and look for resource leveling opportunities. If a more aggressive schedule is needed, the PMI identifies two main ways to compress the schedule:

- Crashing—adding resources to activities to make them go faster. This is not always doable and will typically increase the overall cost of the project.
- Fast tracking—running activities in parallel that were not intended to be in parallel. This can cause mistakes and rework, and typically adds to the risk of the project.

The dependencies and constraints that are discovered in the analysis phase will often be related to customer availability and sign-off. Once the draft schedule is created, review it with customers, users, and other key stakeholders to get their buy-in to the constraints, assumptions, and dependencies.

4.5.4 Costing

Most business analysts are not responsible for the cost estimation of the analysis phase, but they will often have input into it. Different organizations count different line items as part of the analysis cost. The following is a starting point for cost items that should be considered for the analysis phase:

- Analyst time: This is normally included and tends to be the easiest cost item to identify.
- Customer/user time: This is rarely included, but really should be if there is an interest in getting to the real cost of development.
- Analyzing interfaces (even if done by other groups): This is often the work of business analysts, who are often working for a different department, and their time, although it may be charged for, is rarely budgeted for.
- Analysis and development tools: Hopefully, these tools will be used on many projects so the current project should not have to absorb the whole cost. However, in many organizations, the project making the initial investment is the project that is charged.
- Quality assurance (QA): This is often an overhead function and is normally not charged directly to the project. Note that if the QA group does testing, that puts this in a different light. Then it is likely that there will be charges, even for their involvement in planning for the test.
- Process work: Improvements to development processes, roll-out, and training on processes tend to be an indirect (overhead) charge.

Some organizations are starting to outsource requirements gathering. This is likely to lead to a more competitive environment where the costs of creating the BRD will be scrutinized more than it has been traditionally. Being cost conscious and aware of the financial impact of various approaches will make it more likely that the business analyst will be regarded as more competitive.

4.5.5 Tracking and Reporting

Tracking and status reporting can be difficult in the analysis phase. Because this is the phase that relies most heavily on progressive elaboration, it is often difficult to know how much else will surface down the road. Every time a stakeholder is interviewed, it is likely that more stakeholders will be discovered and that more activities will be added to the task list. There must be a balance among the need to interview more stakeholders, the need to finish the phase, and the risk associated with both. Make sure that all key stakeholders are updated and in sync with the approach and the risks associated with it. If the decision has been made that it will be too costly to interview users in other countries, then that decision must be documented and communicated to all relevant stakeholders. This will allow anyone with strong objections to raise them early on, instead of having to deal with them when it is too late to change direction. If there is a lack of communication from the business analyst and it is discovered that some stakeholders were missed in the decision-making process, there will likely be dissatisfaction. However, if everyone is informed of progress along the way, it will be easier to correct any issues from missed stakeholders. It is the difference between a team's mistake and a business analyst's mistake.

4.5.6 Kickoff Meeting

Once a plan has been created for the analysis phase and the analyst is ready to start capturing requirements, then there is a good opportunity for a kickoff meeting. Kickoff meetings can (and often should) be held at numerous times in the project such as

- When a significant activity is about to start
- When there are a lot of new players on the team
- When the direction of the project has changed for some reason

This kickoff meeting is more traditional in purpose, focusing on the players and processes used in the requirements-gathering effort.

As with all kickoff meetings, the purpose of this one is to get everyone on the same page, making sure that goals and objectives are understood, and to build team morale. This one has a special additional purpose. Because requirements gathering is about to start, this is a great time to get commitment for participation from the different stakeholders. It is a good time to review the overall approaches that will be used to capture the requirements and to make sure that everyone is comfortable with those approaches. These approaches include what elicitation techniques the team intends to use as well as which people will actually be involved with providing requirements to the team. This is also a good time to ask who else should be talked to during the requirements-gathering process. One of the most difficult parts of requirements gathering in a large organization is to find all the right people to talk to.

A well-executed kickoff meeting is started by the project sponsor, and, ideally, some upper-level management from both the customer and development organization. Having the right people in the room will show the organization's commitment and will make it much easier to get the users to actually show up for the sessions, fill out the surveys, and participate in brainstorming sessions. Also allow for some team-building time. Because requirements gathering is largely a communications process, it is important that the people involved are as relaxed and comfortable with the rest of the team as possible. Do some introductions, get some ideas about people's backgrounds, and, if possible, set aside some time for socializing. Many projects today are global in nature and consist of virtual teams. It is still recommended to have the meeting in person whenever possible; however, if that cannot be done, a virtual meeting can be a decent substitute.

Having the right people in the room will show the organization's commitment.

4.5.7 Summary

Although the project manager is the owner of overall project planning, the business analyst should be responsible or at least heavily involved with the planning of the analysis phase. The plan created may be simple for small projects or highly complex

for large projects. The plan created by the business analyst should be integrated and really be a subset of the overall project plan. Thus, using the same templates, tools, and processes as the main project plan makes sense. Also make sure that the analyst's plan is reviewed and approved by the project manager before presenting it to the customer.

Some of the requirements plan will actually be used after the analysis phase. Such aspects as product risk assessments will be valid concerns to monitor for the duration of the project.

Activity

Refer to the Swede-Mart case study in [Chapter 10](#).

1. Determine how the three main approaches discussed in this chapter can be used for development of the order system. Using the table below, identify the risks with each of the three major approaches discussed and identify risk mitigation strategies for each. What approach will you recommend for Swede-Mart? (It may very well be a recommendation that contains elements of all three approaches.)

APPROACH	RISKS	MITIGATIONS
Waterfall		
Iterative		
Agile		

2. The project manager has just met with you, the business analyst, and asked you to come up with your plan for the analysis phase. Review the list below to determine what still must be developed. Remember that the focus is the gathering of requirements, not the whole project.
 - a. Project overview and background.
 - b. Scope and deliverables: Identify three examples of exclusions.
 - c. Stakeholder analysis: Fill out the stakeholder analysis for Joe Jones.
 - d. Communications plan: Fill out with two sample communication items.
 - e. Project activities: Create a WBS with three layers (the top layer being the analysis phase) and a total of fifteen to eighteen activities at the lowest level.
 - f. Roles and responsibilities: Assign responsibilities to the activities from activity (e).
 - g. Resource plan: Create a list of potential nonhuman resources and subject-matter experts you may need.
 - h. Requirements risk plan: Brainstorm four to six risks. Select the top one and create a risk-handling plan.
 - i. Identify who will be responsible for approving changes.

5

ELICITATION AND COLLABORATION

Seek first to understand, then to be understood.

Stephen Covey

This chapter will cover primarily what the International Institute of Business Analysis™ (IIBA™) refers to as requirements elicitation and collaboration. While much of the business analyst's work is structured and well defined, and really is centered on discipline, this portion of the job is more an art than a science. A business analyst needs very good communications skills, a strong ability to deal with different personalities, and good facilitation skills. The topic of this chapter, the capturing of requirements, is where these skills are put to the test. The first part of the chapter discusses some of the drivers behind the process of selecting the appropriate elicitation techniques for a project. The chapter then presents a summary review of all the techniques covered, followed by a detailed exploration of each technique. The techniques covered are a subset of the techniques from the *Guide to the Business Analysis Body of Knowledge® (BABOK®)* version 3.

Communicating outputs from the requirements-gathering effort is only a portion of the overall communication with which the business analyst is involved. However, it is an aspect of communication that is often overlooked from a planning standpoint. It includes presenting and making the information available, but also validating and gaining acceptance of the requirements package. Even though the same basic information will be communicated to all stakeholders, the communication needs to be customized to fit the interest and knowledge of each stakeholder.

5.1 Objectives

- Identify the drivers behind selecting an appropriate elicitation technique.
- Determine the best requirements-gathering technique based on the customer's situation.
- Identify different techniques for gathering requirements.
- Describe best practices for each technique.
- Evaluate the pros and cons of each technique.

5.2 Overview

The goal of the business analyst is to be a communications link between the customer and the developer. To do that, the business analyst must find the right elicitation technique. When we use the word *elicit*, we mean to capture, discover, uncover, or simply just write down the requirements. It does not mean invent, create, or guess. Sometimes, it may be as easy as sitting down to interview a customer; other times, it may involve a facilitated session attended by people from all over the world, maybe even using videoconferencing. Each situation is unique, so part of the upfront work is to identify which technique (or techniques) best suits a specific project. Is this a survey project? How about job shadowing? Will this customer work well in a facilitated session? Is there political tension in the organization that will make one approach more desirable than another? Each of them has pros and cons, and can be useful at times.

Each situation is unique, so part of the upfront work is to identify which technique (or techniques) best suits a specific project.

The elicitation techniques presented in this chapter are from IIBA-defined approaches, industry best practices, and other approaches that I have encountered over the years. Although there may be some techniques that are not covered here, they are likely to be a close approximation to one that is covered. It is by no means intended to suggest that the business analyst choose only one of these techniques. Rather, view these techniques and approaches as tools for the business analyst toolbox. There is enough material here to assist with creating a customized plan for any project by combining and customizing techniques.

Table 5.1 shows an overview of the techniques covered in this chapter and some of their main uses in the analysis phase. Notice that there are many overlaps in the uses of the techniques. Use the guidelines from Section 5.4 to identify which techniques may work best. In this chapter, the techniques will be explored to determine when to use them and to identify some best practices for them.

5.3 IIBA View

The IIBA has identified five tasks that fall under elicitation and collaboration:

1. Prepare for elicitation.
2. Conduct elicitation activity.
3. Confirm elicitation results.
4. Communicate business analysis information.
5. Manage stakeholder collaboration.

Table 5.1 Requirements-Gathering Techniques

TECHNIQUE	WHEN USED	COMMENT
Customer interviews	All situations	Often a subset of other techniques
Observation/job shadowing	User requirements Business analyst new to environment User not experienced with systems	Requires time commitment from both business analyst and user
Studying existing systems	Understand as-is Prep work for customer interview Technology change with old business processes	Can be time consuming Documentation must be available
Studying interfaces	Understand big picture Understand project impact	May involve people outside of the project team
Surveys	Wanting to reach a large population Geographically disbursed population Wanting consistency in interviewing	Realistic expectations on return rates
Discovery/JAD/facilitated sessions	Multi-stakeholder environment Consensus building is needed	Large undertaking, requires formal planning
Focus groups	Gather a lot of ideas without committing to implementation Early phase of project	Set realistic expectations
Market research	Early research Evaluate what is available on the market	May limit creativity
Evaluate best practices	Early research Evaluate what the competition is doing	Information may be difficult to find
Prototyping	Reduce abstractness Bridge language gaps Do throughout requirements process	Use with other techniques from this list
Storyboarding	Speak customers' language Reduce abstractness Do throughout requirements process	Subset of prototype but no functionality
Idea-generating techniques (brainstorming)	Early stages Gather large number of ideas	Must keep focus

The best practices of these techniques are covered later in this chapter, but here is an overview.

1. Prepare for elicitation. The type of preparation needed will vary from technique to technique but some examples of activities that should be considered during this task are
 - Establishing ground rules
 - Getting agreements from the participating stakeholders
 - Determining any potential training needs of the participants
2. Conduct elicitation activity. This is, of course, where the actual elicitation takes place. The technique selected and prepared is carried out and the output is documented. Each technique has its own set of challenges for the person responsible for the elicitation, which may be the business analyst, a facilitator,

or some other designated party. Elicitation results must be documented. This may be done on whiteboards, with audio or video recording, or most commonly, in writing. The format of the documentation should be predetermined and agreed to by the stakeholders. Different elicitation techniques lend themselves to different types of documentation, and care should be given to make the documentation as easy to produce and as useful as possible.

3. Confirm elicitation results. Once the requirements are captured and documented, they should be reviewed to ensure that the business analyst's understanding of the requirements matches the stakeholder's needs.
4. Communicate business analysis information. The information must be structured and presented in a way that fits different stakeholders. A communication package needs to be prepared for each group and reviewed with that group in the most appropriate manner.
5. Manage stakeholder collaboration. Stakeholders must be engaged. When multiple stakeholders are engaged, the effort must be on keeping them focused and interested. The business analyst is involved with making sure that the agreements and commitments are kept.

5.4 How to Select the Right Technique for Gathering Requirements

What drives the selection of one technique versus another for a project? Although it is often just a matter of what the analyst is comfortable with, it actually should be driven by a situational analysis. The three main drivers explored in this section are

- Customer
- Category
- Geography

There are others as well, though. Existing skill sets, past experiences, preference of decision makers, and just plain intuition are also factors to be considered. The selection of the best technique is an important decision. If the project manager decides to do customer interviews and then finds that the *customer* is a diverse group with diverging opinions about what the requirements are, the whole project may fail because of it.

For example, to look at the Prescription Interaction Project again, the pharmacist will have requirements for the system that will deal with usability, the customers (of the pharmacy) are likely to have privacy concerns, and management will focus on productivity and minimizing legal liabilities. If all of these stakeholders are interviewed separately, these conflicting requirements are not likely to be raised and resolved. However, if there is a facilitated session with all three parties represented, the discussion will likely bring up these differences and they can be dealt with right in the session as they are identified.

5.4.1 Different Ways for Different Customers

The biggest reason for selecting an elicitation technique is the customer. There is a set of start-up questions that must be analyzed with regard to the customer:

- Is this a single or multi-stakeholder environment? A multi-stakeholder environment is the hardest one to manage, especially where there is some level of controversy or disagreement concerning the requirements. When stakeholders from different organizations have conflicting, and perhaps valid, opinions about what a product should do, the role of the business analyst becomes much more difficult. It is not the analyst's job to make the decisions or to decide on the *right* requirement. However, it is the analyst's job to create an environment in which stakeholder prioritization will lead to a consensus on how to go about decision making in this scenario. It will take strong facilitation, extensive communication, and a lot of flexibility to be successful. In the end, it must be the customer who makes the decision.
- Are all the customers located together, or are they spread out geographically? If they are spread out over North America, or over the globe, it will take more creativity to get the requirements. Travel costs, the limits of technology, and the priority of stakeholders all must be used to evaluate the best technique. Although a face-to-face interview may often be the preferred approach, in reality phone conferences and e-mails may be more practical.
- Is this a well-understood business environment, or is this brand new to the customer? If it is well understood, there will be less need for discovering the requirements and more for just documenting them.

5.4.2 Different Ways for Different Categories

When working with executives on business requirements, there is typically a need to be more conservative with the time they need to spend with the project as well as a need to be flexible with scheduling. When working at the user requirements level, there may be more need to actually spend time with the customers while they are performing their work, or to show them different potential scenarios in a prototype. When exploring regulatory requirements, the focus tends to be on reviewing existing documents and talking to subject-matter experts (SMEs). The requirements-gathering approach for each category should be reviewed independently and evaluated for the best approach. Even though interviewing may be the best approach for the executives of the organization, that approach may not work at all for the end users. Looking at the Prescription Interaction Project, there may be a need for interviewing executives, surveying doctors, and job shadowing of pharmacists.

5.4.3 Impact of Globalization

In recent years, globalization has had more and more impact on requirements gathering, at multiple levels. First, there is the obvious one, with users and business organizations located in different countries, speaking different languages. Then, there is the development aspect. When the intention is for the system to be developed in a different part of the world than where the customer is located, there must be special attention not only to getting the right requirements in the right fashion, but also to making sure that those requirements are documented in a clear, concise, and understandable manner. That may increase the need to use prototypes and review sessions including both customers and developers.

There are three obvious concerns with a global team:

1. Language—Although this can be a sensitive topic, it must be thought through. If some of the customers or developers do not speak (or read or write) English (or whatever the primary project language is) very well, this must be addressed early on in the project. It may involve translators, selecting different team members, or just selecting a different approach to communicating. To avoid the misunderstandings of verbal communication, there should be a stronger focus on models, prototypes, and written communication. Language difficulties, where they exist, should be documented in the risk list, and plans for how to deal with them must be developed. It is a risk that is tempting to underestimate because it is often emotional to bring it up. It should be remembered that no one benefits from communication issues: not the project, not the customer, and most definitely not the person with the communication problem. Is this a risk that should be handled by the project manager or the business analyst? It could actually be done by either. The project manager looks at potential risks to the overall project and may very well identify this risk. However, because the business analyst is responsible for creating the business requirements document (BRD), he or she may identify this as an analysis phase risk. Either way, the project manager and the business analyst must share this type of information with each other.
2. Culture—Much has been said about different cultures around the world and the impact they have on communication styles. Steamrolling Americans, team-focused Europeans, and nonargumentative Asians are stereotypes that may be true in some cases; however, it is much more important to focus on the people involved and their personalities rather than a whole culture. That said, when gathering requirements from a group of people, it is important to understand the mode of communicating that is customary in their environment. In a discovery session in Asia, there may be a tendency for people to appear to agree with the stakeholder, while in reality all they are showing is that they are hearing and understanding the discussion. In such cases, there may need to be one-on-one sessions to further explore and get buy-in. Remember,

though, that individual differences are often great and that a specific customer in the United States may very well have a more Asian or European approach to communication. So, be aware of differences, learn about cultures, but do not assume that every person in a culture acts the same.

3. Time zones—This is sometimes the most obvious issue when dealing with overseas projects, but it is also often overlooked. It is very difficult to get people together and fully focused in the middle of the night. Some of this can be overcome by doing more written communication and by having each organization alternate who attends the meetings as well as by varying the timing of the meetings. Remember that just because the contact person located on the other side of the globe is attending the phone conference at 2:00 a.m. local time and seems to be on the same wavelength as the rest of the meeting participants, it does not mean that the rest of the team on the other side of the globe will get the same message. What often happens is that there is another layer of communication added where the contact person acts as a go-between who must then bring that team up to speed. If there are further questions or if the contact person missed an important detail, the team may be heading down the wrong path. One good *best practice* is to follow up with a detailed, written account of what was discussed in the meeting. Toward the end of this chapter, there will be a discussion about some of the tools that can help with communications to virtual teams.

One good *best practice* is to follow up with a detailed, written account of what was discussed in the meeting.

5.5 Customer Interviews

Although customer interviews can be a stand-alone technique for gathering requirements, it should be noted that most of the other techniques covered in this chapter also contain interviews in some form. So much of what is covered during the interview discussion is also applicable to job shadowing, surveys, and discovery sessions.

The basic, traditional interviewing approach is to identify key stakeholders, interview each of them, and then analyze the results from all the interviews. This technique works well when there are only a few customers and little disagreement expected among them. It is also a necessary approach when there is a geographically dispersed team, meaning it is difficult to pull the whole team together, or when dealing with stakeholders who have very little time to set aside. When the main approach is one-on-one interviewing, it is important to make sure that the requirements are shared with all stakeholders after the interviews are conducted. This will assist in obtaining buy-in and will allow potential disagreement and differences in assumptions to come to the surface. When customers give their requirements in an interview, they will typically state what they think are most important for the product, from their

viewpoint. They will leave out other ideas, not because they do not want to do a good job, but rather because they assume that the analyst already knows the information. Different customers will make different assumptions, and by verifying and comparing different customers' requirements, the analyst can identify conflicting assumptions and requirements. Start this verification process by compiling all the requirements in one document and sending it out to the whole population that provided input. If there is a concern that they may not actually read this document, consider meeting again with them one-on-one or conducting a formal walk-through of the document.

Different customers will make different assumptions, and by verifying and comparing different customers' requirements, the analyst can identify conflicting assumptions and requirements.

There are two main types of interviews. First, there is the open-ended interview, where the interviewer starts with a list of key questions, and then expands on them based on the answers from the interviewee. This is the most common approach and it works well, especially when there are few stakeholders and the interviewer is experienced. If there are many stakeholders or if the interviewer is less experienced, the second approach, a survey interview, often works better. In this scenario, the interviewer predefines all the questions and goes through the same list of questions with each stakeholder. It is very similar to the written survey, which will be discussed later in this chapter, with the main difference being that it is a live session so there is a chance to probe into areas that may not be clear.

Interviews, though common, are sometimes the most difficult requirements-gathering technique to perform properly. There is a tendency to view the interview as an informal approach and, as a result, not do enough preparation or not capture the result from the session in a formal enough manner. This is especially true when the customer is someone that the analyst has worked with in the past and with whom there have been many previous meetings. When using interviewing as a requirements-gathering technique, it is better to use a formal and repeatable process. That will increase the customer's understanding that this is more than just small talk, and it will ensure (or at least improve the odds of) a productive session. The following steps are a good approach to interview preparation:

1. Determine the high-level need for an interview. This should come out of the stakeholder analysis discussed in [Chapter 4](#). Because there may be many people to interview and there is often a need to select whom to include in the interviews, this initial selection must be done as early as possible. Make sure to review this with the sponsor and the project manager; they may provide insight on key stakeholders who should be included. The analyst tends to select interviewees on the basis of the quality of the information they can provide. The project manager and the sponsor can often point out people who

need to be included for political reasons, which may be just as important. In most organizations, there are different types of power. The person who has the power on the organization chart is not necessarily the person who will bless the requirements. There are often SME(s) within the organization, the person or persons to whom the official decision maker will go in order to review the requirements. Always try to identify these *key users* or *super users* within each organization. These are the users who know the most about the current environment. If they do not buy into the requirements and the final solution does not meet with their approval, the whole effort is likely to fail. They can be very rooted in the current environment and may not be willing to accept significant change. Typically, they have been successful in the current environment and may feel threatened by any drastic change. The business analyst should strive for inclusion of both these experts and of people with less experience in the current environment. After all, the goal is to capture good requirements for the future, rather than to document a future that is a personal agenda of one customer.

2. Schedule the interview with the customer. At this time, all the details about the interview are yet to be worked out; however, there is at least a slot reserved on the customers' calendars. Let the customers know that more information regarding the interview will be forthcoming, but that the purpose of this notice is to give as much lead time as possible. Always show the customers that the analyst realizes that they have a regular job to do, which is important, and that there is a high level of respect for their time commitments.
3. Do the homework. Research the customers' areas of knowledge and determine what the boundaries of their involvement should be. Review existing documentation (if any), for both the business area and any current systems. If this customer is not known to the interviewer, also spend some time finding out more about the interviewee: talk to other analysts, the project manager, or the sponsor. It is good to understand the personality of the person being interviewed, as well as any personal or professional biases. It is the job of the analyst to distill requirements and to try to remove personal preferences. Sometimes, this homework will actually lead to a change in the requirements-gathering approach. On the basis of understanding the personalities involved, there may be a better technique to gather the requirements (such as job shadowing, which will be covered a little later in this chapter).
4. Draft an initial list of questions. It always leaves a good impression when the interviewer can show that time has been spent preparing for the interview. It also provides a safety net if the conversation runs dry. Most important, it ensures that the right questions are being asked. It is easy to get distracted and forget what the intentions were going into the interview. A sample question list has been included in [Figure 5.1](#). Review this list of questions and then

- What do you expect this project to accomplish for you?
- What other projects are helping you achieve these goals?
- What organizations within your company will be affected by this?
- Have you attempted this, or a similar, project in the past?
- Do you have a clear image of what the end product should look like?
- What should the system do?
- With what business areas must the system interface?
- What are your schedule and cost requirements?
- Is schedule or cost most important for you (or are they equal)?
- Have you seen a product like this that you liked?
- How do you accomplish these goals today?
- How will you evaluate the end product?
- What are your criteria for success for this product?
- Does this project have the support of your management?
- What are your competitors doing for these types of functions?
- Are there any government regulations of which we should be aware?
- Is there a union impact?
- Are there interface requirements to other parts of the business?
- Are any other projects dependent on this project?
- Are there any other projects upon which we are dependent?
- To whom else should we talk?
- Who is the most knowledgeable person on this topic?
- Are there any security issues?
- Are there technical standards that we have to meet?
- Are there any training needs?
- Are there penalties for being late, rewards for being early?
- Are there industry standards concerning this product?
- Who is the primary user?
- Who else will use the system?
- Do you have specific performance requirements?
- How will change requests be managed?
- Are there any potential liability issues?

Figure 5.1 Sample interview questions.

augment it with questions specific to the current project. Some examples to add for the Prescription Interaction Project are

- Who can enter a prescription?
- How long does it take to fill a prescription?
- What are the customers' responsibilities in this process?

5. Update the customers on the details of the interview and provide the list of questions ahead of time. Even if the customers do not read them, it sends the message that this is something that the analyst is preparing for and that it should be taken seriously.

When planning for the session, there is a natural flow of questions that should be followed. There are four main types of questions that the interviewer will use:

1. Open-ended ("Tell me about"). This allows customers to open up and be descriptive about what they do in their environment. It also allows them to introduce topics of which the interviewer may not have been aware. Some

customers can become overly verbose here and lose focus; others may not have much to say at all. If either of these situations is the case, it may be better to use closed-ended questions, which will be discussed next. Examples for the Prescription Interaction Project are

- What do you do with a prescription when the customer gives it to you?
 - What are some of the difficulties with the current process?
 - What would an ideal system look like to you?
2. Closed-ended (“Do you enter orders into a system every day?”). These questions require short answers, sometimes yes or no, and are intended to keep customers focused and to let them know what type of information is needed. Some closed-ended questions can be prepared ahead of time, but many of them will come from responses to questions asked earlier in the interview. Identifying these additional questions throughout the interview takes both experience and active listening on the part of the interviewer. Examples are
- How many customers in a day will fill a prescription?
 - Of those customers, how many have drug interaction problems?
 - What are the peak hours at the pharmacy?
3. Probing (“What do you mean by ‘zero defects?’”). These types of questions are used when there is a need to clarify what the interviewee means by a statement. There will always be some assumptions involved with any documentation, but there should be an effort to minimize them. For example, the customer may state that the system must be available during regular working hours. Some examples of good probing questions are
- Are your peak hours consistent throughout the year?
 - Do all pharmacies have the same open hours?
 - What does the customer do to fill a prescription when you are closed?
 - Do you plan to expand into overseas markets?
4. Validating (“Here are the requirements that have been captured thus far. Are they correct?”). An interview can be a lengthy process, and the discussion may have taken many turns along the way. Stop at times and make sure that there is agreement on what has been said thus far. Because each step builds on the previous responses, any invalid information captured will be with you for the rest of the process. If the documentation states “all print outs will be in English,” when the customer really meant “all internal print outs will be in English; all external customer print outs must be customizable,” then that difference can result in drastic changes to the overall project. Examples of validating questions are
- Does this flowchart accurately reflect your prescription fulfillment process?
 - I heard you say that the customers often use relatives’ prescriptions, which is a problem. Is that correct?

In general, the flow through these questions would be in the sequence shown: starting with open-ended questions, narrowing down answers with closed-ended

questions, clarifying with probing questions, and getting commitment with validating questions.

The interviewer should follow a standard, repeatable process for the interview itself. Although a good business analyst must have strong communication skills, those skills should not be confused with the ability to make small talk. Naturally, the more personable the interviewer is, the more relaxed the interviewee tends to get. This is good, but make sure that there is an agenda for the meeting and that the goals and objectives for the interview are known to both parties. [Figure 5.2](#) shows a sample agenda for a customer interview that can be customized to fit the situation.

Make sure that there is an agenda for the meeting and that the goals and objectives for the interview are known to both parties.

Start the meeting by ensuring that the interviewee knows what is going on. If the parties are not familiar with each other, then start the session with introductions and backgrounds. It is important to be conscious of the interviewee's time commitment and not go into a lengthy life history; however, the rest of the session will go much smoother if good rapport is established up front. If possible and the interviewee is responsive to the idea, get the questions to him or her ahead of time. That allows time for them both to consider the questions before the interview and to check with other people in their departments.

Also spend some time learning about the interviewee's background and level of expertise. This is important for a number of reasons. First, it shows an interest in the person and that the expertise that he or she brings to the session is valued; second, it will bring out useful information needed to ask the right questions and determine the right way to document and model the results of the session. If the interviewee has little experience and knowledge in the systems development area, there must be a

Interview agenda: Prescription tracking project

Stakeholder: Diane Walgreen

Interviewer: Bonnie Smith

Date: January 4, 2007

- Introductions
- Establishing boundaries
- Current situation—defining problem
- Identify problem cause
- Envisioning the solution
- Requirements of the solution
- Recap and validation
- Next steps

Figure 5.2 Sample interview agenda.

much stronger focus on validation of the requirements, and the models used to document the requirements would mostly be simplistic and intuitive in nature, such as a workflow diagram. However, if interviewees have been through this process many times, they are more likely to know what type of information is needed, and they are often more familiar with more advanced modeling techniques, such as data models or use cases. There is a more in-depth review of modeling techniques in [Chapter 8](#).

Two analysts should attend each interview. This will allow one to focus on scribing and one to focus on questioning. They can take turns and tag team, which will be less stressful. It also allows for two views on what the customer actually said and avoids potential misinterpretations.

After introductions, it is time to start building the interview step by step, following the concept of starting with what is known and then gradually moving into the unknown. There are five main steps for the analyst to follow:

1. Understand the boundaries. The analyst should already have a good understanding of the boundaries of the project at this point. The interviewee needs to be brought up to the same level of understanding. In addition, it needs to be clear what the boundaries of this specific interview are. Although the project may be focused on prescription tracking, this interview may only be dealing with drug interactions. [Figure 5.3](#) shows an example of using a functional decomposition diagram (FDD) to show the boundaries of an interview. In

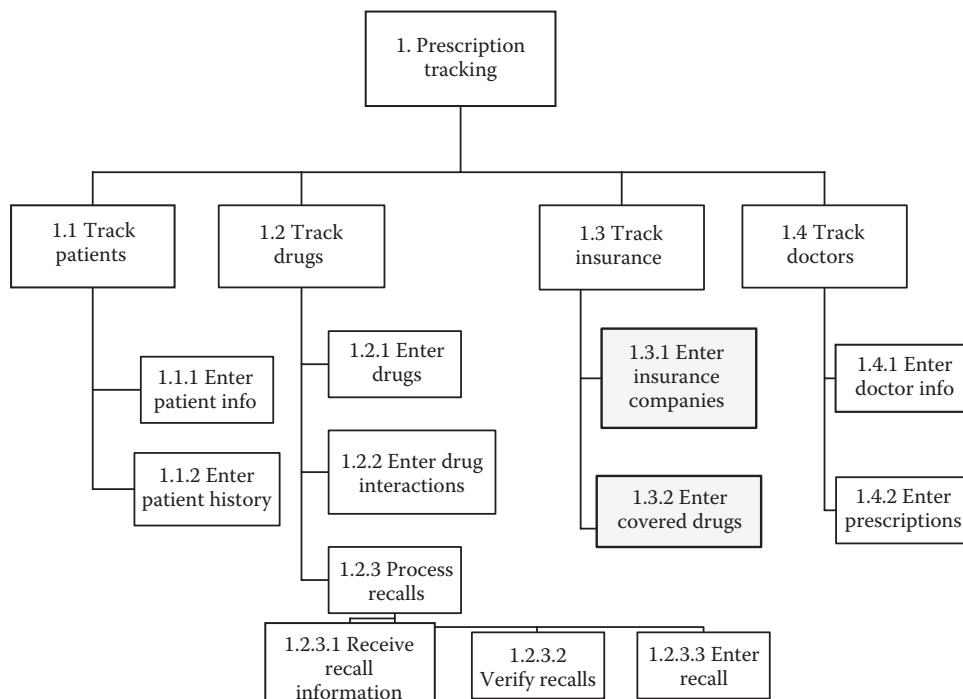


Figure 5.3 Establishing interview boundaries through the FDD.

this example, the interview is focusing on process “1.3.1 Enter insurance companies” and process “1.3.2 Enter covered drugs.” By clearly establishing this up front, the analyst and the stakeholder can stay on track and avoid lengthy discussion on topics outside the objectives of the session. The FDD itself and how it is constructed will be explained further in [Chapter 8](#).

2. Define the problem. Projects are typically done in response to a problem. It can be a regulatory problem or a business-driven problem, but either way the problem must be defined so that the stakeholders can see why it needs to be fixed. If the problem is stated as “Pharmacist cannot see prescriptions filled at other pharmacies,” then it may be viewed as a problem, but not necessarily a critical one. On the other hand, if it is stated as “Pharmacy is being sued over drug interaction problems,” then that is a more urgent-sounding problem. State the problem in terms of the real impact. The project problem should have been stated up front; here, the focus is on stating what the problem is from this stakeholder’s perspective.
3. Identify the root cause. There are often multiple causes for a problem. The main root cause may not be within the domains of this stakeholder, but there should be a review of what the stakeholder views as the root cause. There should also be documentation of root causes that have been discovered but not addressed by this project. The problem statement “Pharmacy is being sued over drug interaction problems” can be caused by
 - a. Poor tracking of prescription drugs within our pharmacy
 - b. No information sharing with other pharmacies
 - c. Patient buying drugs overseas
 - d. Patient getting drugs from relatives

This project may deal with (a) and (b), but not with (c) and (d). Because only part of the problem will be solved, this should be documented and communicated.

4. Envision the future. As an analyst, there is a tendency to get into the details of a potential solution too quickly. Because the deliverable will be a BRD that must be detailed enough for the systems developers, there is a rush to move from the fluffy to the precise. Instead, spend some time exploring what the stakeholders would envision their jobs and environments being like after the problem has been solved. Envisioning the future provides context and reality to the actual requirements. It helps transition the conversation from the *as-is* situation (the problem and root causes) into a *to-be* discussion (vision and requirements of the solution).
5. Determine requirements of the solution. The focus here is on what this stakeholder needs to make the vision a reality. Not *how* it will be done, but rather *what* must be done. This is true whether this is an interview with a core end user or this is defining an interface with a different system. In either case, the focus is on capturing the information that the interviewed party or organization needs to be successful in the new *envisioned* environment.

Advantages of interviews:

- They are easy to schedule because there are normally only two or three people involved with the session.
- They can be more informal, which may work best with some stakeholders.
- They are flexible, meaning they can be rescheduled if conflicts arise, and they can go off into unplanned directions, allowing for more exploration of different issues.
- They allow for probing into more details than some other techniques. If an answer is not clear or an inconsistency is found in the answers, it is easier to question and explore that when there is only one customer in the room.

Disadvantages of interviews:

- The customer often feels less committed to an interview. If a conflict arises, the customer often cancels the interview or cuts it short.
- It is more difficult to control interruptions in an interview than in a group setting. Customers often will accept phone calls or even allow interruptions by other people in the middle of the interview. Try to get their dedicated times and meet in an area other than their offices.
- Interviews tend to be time consuming (especially in a multi-user environment). There is a lot of information that needs to be revisited with each person being interviewed, and there is often a need to go back to someone who has already been interviewed, on the basis of information discovered in later interviews.

Best practices:

- Go into the interview prepared. Have a list of questions and share it with the interviewee ahead of time.
- Have two analysts in each interview.

5.6 Observation or Job Shadowing

Observation or job shadowing can be done actively (interacting with the user) or passively (using video recordings or one-way mirrors). This technique is used primarily with the end users of the system being developed, most likely the pharmacist in the example used here. It can be a very effective technique when one is trying to understand what the user is doing on a daily basis, because it is often difficult for a user to spend a lot of time explaining the business and the system needs to the analyst. This approach can be effective because just talking about business needs and requirements without a solid frame of reference can be too abstract. Often users do things without thinking about the details of what they do. Job shadowing allows the business analyst to see the work actually being done, rather than hearing what the users think they are doing (or what

they are able to explain to the analyst). A much more productive result can be achieved when the analyst can work alongside the pharmacist and ask questions. However, being observed might also alter the pharmacist's behavior, so that must be taken into consideration (see the discussion on the Hawthorne effect later in this section).

There needs to be careful consideration of the method of observation used as well as the subjects of the observation. Most people are a bit uncomfortable with the thought of being observed while they are working, and there is often a concern about what the observation results will be used for (sometimes rightfully so). The main methods of observation are

- Job shadowing—One-on-one interaction between the analyst and the user. The analyst sits (or walks) together with the users, watches what they are doing, and asks questions (either in real time, if the situation allows, or at the end of the day, if that is more suitable).
- Recording—The user is recorded while performing tasks. The recordings are then edited to get a variety of situations and users to analyze. This can compress the time frame needed to analyze data and may work better in some customer environments.
- One-way-mirror-type setups—The analyst observes the user in real time, but the user does not see the analyst (however, the users normally know that they are being observed). This can allow for observing multiple users at the same time.

Regardless of which method of observation is used, there must be a strong emphasis on building trust with the person being observed. There needs to be a clear understanding of what is being observed, why it is being observed, and for what the data will be used. When gathering information for systems development, the data gathered must not be used for evaluation of individual job performance. If it is used for that purpose, then the results will inevitably be skewed and unreliable.

While observation is a great technique to increase the analyst's understanding of what the users actually do in their work, it is also a technique that can be very time consuming and difficult to limit to what is important for a specific project. Because the purpose is to observe what the user currently is doing, it is primarily an as-is situation that is obtained using this technique. If the intent is to drastically change the way the customer does business, then there may be limited value in the results achieved. Another natural limitation of the technique is that users are observed doing their whole jobs. Normally, what the analyst is interested in is only a subset of a user's job, so too much time may be spent on areas outside the scope of the project. It is also likely that most of the effort will be spent observing the normal behavior (the way that business is normally done), while there are probably many exception scenarios that the analyst would never see during the time that the observation takes place.

So, with all this said, should observation be avoided? No. For many stakeholders, it is a great way of capturing their requirements; however, be aware of the drawbacks and compensate for them. Supplement the observation results with closed-ended,

probing, and validating questions. The observation will give a good base of what the user needs. The interviews and discovery sessions will bring it to the next level.

When planning for observations, the Hawthorne effect must be considered. The Hawthorne effect was named after a plant in Cicero, Illinois, where Western Electric conducted a productivity study in the 1920s. They changed the lighting in the plant and then studied the impact on productivity. For some workers, they increased the lighting, for some they kept it the same, and for some they decreased the lighting. The surprising result was that productivity increased for all three situations. The productivity increase had less to do with the lighting than the fact that people were walking around in the plant observing and writing down what people actually did. Most people focus more on the task, take shorter breaks, and waste less time by the water cooler when they are being observed. So be aware that observing the users will actually change the way they do their jobs.

Advantages of observation:

- It allows analysts to see what users actually do, not what they think they do (or would like the analysts to think they do).
- A number of users can be observed at the same time to see what the differences in individual practices are.
- It removes the need to take the user away from the work environment.
- It is less abstract than interviewing because the actual work being done is observed.

Disadvantages of observation:

- Sometimes, it is difficult to discern reasons for behavior. The analyst can see what the user does but may not know why.
- People act differently when observed.
- A lot of time may be wasted watching the users do activities outside of the project boundaries.
- Recording and documentation may be difficult.

Best practices:

- Make sure all participants know why they are being observed, what is being observed, and what the observation data will be used for.
- Complement observation with interviews and discovery sessions.

5.7 Studying Existing Systems

If the analyst goes straight to the customer and starts asking for requirements without first evaluating the existing systems, then it is likely that the customer will get frustrated. The customer will expect that the analyst has done some homework and has an understanding of the as-is environment, especially from a systems viewpoint. Another

reason to study existing systems is when the main purpose of the project is to introduce new technology. In other words, the project is driven by a technology change, not by a business change. Customers may not feel the need to meet and discuss requirements because, from their viewpoint, they just want things to work in the future the same as they work today. Although it is a good starting point to review existing system documentation to know what currently exists, it is also important to realize that the systems in use today may no longer be used as originally intended and documented, and the users may have many workarounds for things not functioning well. If those areas are not identified up front, it is likely that the customer will get a new and pretty system with the same problems as the old system. This is sometimes a valid approach, especially when investing in a new platform; however, it should be a conscious decision, and the customer should have the right expectations going into the project.

Examples of artifacts to review to get an understanding of the current situation are

- System documentation
- Enhancement requests
- Problem logs
- User manuals
- Training manuals
- Product literature (both for the existing product and for competing products)

Because this can be a very time-consuming effort, the analyst needs to make some judgments as to which documents are valuable, which ones are outdated, and how much time and effort is worth spending on this activity.

Advantages of studying existing systems:

- It limits customer involvement.
- It might give the project a jumpstart with a lot of existing documentation.
- There may be valid requirements built into today's systems that the users would not think of mentioning.

Disadvantages of studying existing systems:

- It may not have updated documentation.
- It may not reflect the way business is done.
- It may be time consuming for little return on investment.

Best practices:

- Determine how relevant the current system is to the future functionality.
- Determine the accuracy of documentation.

5.8 Studying Interfaces

The IIBA groups user interfaces with other interfaces when discussing interface analysis; however, in this section, the reference is to system interfaces. User interfaces such

as prototyping and storyboarding are discussed in other sections of this chapter, and in [Chapter 8](#), which covers modeling.

Most systems being developed and used today will need to interface with other systems. Even though it is possible that a system is self-contained, with a user entering, analyzing, and reporting on data in the system, that scenario is uncommon. Instead, a system typically is dependent on other systems for data and processing of information, and other systems may be dependent on data and functionality from the system being developed.

Identifying the necessary interfaces can be a complicated process. Many of them can be identified by reviewing existing systems, analyzing where their data are coming from and what triggers them, as well as identifying where information is being sent to other areas. If there is a current system in place and if the business process is not intended to change much, then most interfaces can be discovered this way.

However, if the business process is changing or if more of the business process is being automated, then identifying the needed interfaces becomes more important as well as more difficult. This analysis needs a well-performed enterprise analysis (as discussed in [Chapter 7](#)) to be able to see how this system fits in with the rest of the organization. It is typically not enough to just ask the immediate customer. Although customers probably can explain what other areas and systems they are dependent on, they may not realize who in the organization is dependent on them.

If the business process is changing or if more of the business process is being automated, then identifying the needed interfaces becomes more important as well as more difficult.

In addition, system interfaces may be beyond the knowledge or even the interest of the immediate customer. A system interface may consist of two different systems sharing a customer database. Each set of customers may view it as their own database; however, from an organizational view, it is preferable to standardize on one common set of data. Processes can have similar interfaces. If the system needs to perform a credit check, and there is another system already doing that, it would make sense to use the same common functionality, which is really a form of an interface. Again, while valuable in the long term, the customer may not see the connection when discussing the project, so the developers and the business analyst must identify these types of interfaces.

Often in projects, the interface analysis is more of an afterthought. It is on the task list, but is relegated to the end of the requirements phase. This is a mistake! System interfaces are really constraints for the product being developed, and constraints must be identified as early as possible. If the product being developed must use the corporate customer database, then that should be considered when conducting Joint Application Development® (JAD®) sessions and developing prototypes. If not, the customer may either develop the wrong set of expectations or be spending time defining functionality that has already been defined.

Another reason to identify the interfaces early on is because they may need to change. If the efforts of the current project require new or updated interfaces to other products, someone will need to make those modifications. Often those modifications are beyond the estimates done for the project, meaning there is also an additional financial consideration for that effort. Even if that can be resolved, the resources needed to make the changes are possibly from a different organization, with different priorities, which may require getting the sponsor involved to help negotiate for priorities. In an ideal world, this all would have been part of the initial development of the project charter and scope statement; however, many interface requirements will be discovered later in the process. When that happens, they should trigger a change control request, and an impact analysis should be performed.

Advantages of studying interfaces:

- There is a need to understand system dependencies.
- It is necessary to fully estimate the cost and impact of the project.

Disadvantages of studying interfaces:

- None; at some level, this must be done for the project to be successful.

Best practices:

- Get existing interfaces from developers.
- Review business process changes to identify new interfaces.
- Look for common data and processing across business areas.

5.9 Surveys

In situations where there are a large number of customers or users, or where logistics make it difficult to get together in person, a survey is a common technique to capture requirements. Surveys are good at reaching large numbers of people in a short time frame and also help the customer stay focused on the topic of discussion. However, surveys also have some clear disadvantages. It is sometimes difficult to understand how to interpret an answer without the ability to ask follow-up questions. There might also be some difficulties getting responses back from busy customers.

Designing a survey is not an easy task. Most of us at some point have seen a survey that seemed poorly designed, too complicated, too lengthy, or just useless. Part of the problem is that when a survey is designed, it is tempting to try to capture as much information in as many areas as possible: since the customer will do the survey anyway, and you have the customer's attention, you want to get as much out of the survey as possible. That is the wrong premise to start with when designing a survey. Instead of asking "What are all the things the project needs to know about?", ask "How much can I realistically get this customer base to give me?" If the survey has 100 questions and they are all complicated and difficult to read, then customers will not return the

survey, or will answer without thinking, or will selectively answer the questions in which they are interested. A much better approach is to pick five to ten key questions and ask them in simple language. With that approach, the survey results may actually be returned and the information provided useful.

Make sure to write the questions in a neutral way. Sometimes, it is easy to see the preference of the analyst come through in the phrasing of the question. A good question should be clear and should not lead the respondent to answer in a particular way. Moreover, the use of negatives should be avoided, as they tend to confuse the reader.

The following decisions need to be made before the survey questions are actually written:

- Who will be the survey participants? If there is a large population to choose from, a random selection can be used. If the population is smaller and there is a need to get a representative level of participation from different customer types, a targeted survey group would be more appropriate. The level of response may also be higher if a targeted group is selected.
- What is the reading level of the survey participants? Part of the user profiling done earlier in the analysis phase (and discussed in [Chapter 3](#) of this book) should identify the levels of education most users have as well as their computer literacy. There is no point sending out a brilliant survey only to find that most survey takers either misunderstood the survey questions or just gave up on the whole survey because the language was too complicated.
- Are there any language concerns? Is the survey population comfortable with English? Should the survey be translated into multiple languages?
- What type of questions will be used (open-ended, rated on a scale, or multiple choice)? Open-ended questions give the survey taker more of a chance to expand on a topic. However, if there is a large population, the evaluation of the survey results will become very time consuming. There is also a tendency for most people to be very brief when answering open-ended questions (or not to answer them at all). The most common survey approach is to have statements with a rating scale indicating how the survey takers agree or disagree with a statement or question. An example: How important is it that the system supports multilingual input? The survey taker can then mark any value on the scale. Most scales have between five and ten options. Some people recommend an even number of options because that makes it impossible to select the neutral, middle choice. Make sure that the survey states which value is good. Some surveys use 1 as the best value; some use 1 as the worst value. It does not really matter which is used, but because 1 is used most often as the lowest rating, it may avoid confusion to use such a scale. The final type of question is multiple choice. This is a good way to present some optional solutions and get users' views on their preferences. This is also a great way of ranking requirements to find out which ones are more important to a large user population.

- How will the survey be returned and who will compile the results? Hidden in this question is one of the most asked questions about surveys. Should they be anonymous? If the survey is not anonymous, it is less likely that the answers will be totally honest. However, if it is anonymous, there is no opportunity to ask follow-up questions and probe for more details. Consider whether the survey questions may be viewed as controversial and make a decision based on that. If it is anonymous, there is also a need to set up a process whereby the survey is returned without anyone being able to trace who turned it in. One way to do that is to have the participants send it to an independent person in the organization who collects the surveys, collates the results, and forwards them to the analyst. If they are not anonymous, they can just be sent to the analyst.

Because there is always a concern that the survey would not be returned in a timely manner or at all, try to involve management in the distribution of the survey. It does help when management shows support for the process and asks the survey takers to make it a priority to fill out and return the survey.

There are survey tools available to help. One that was used to get feedback for this book was surveymonkey.com. It can help with the administration, setup, and evaluation of the surveys. Currently, the website offers a free basic version and, for a fee, there is a more full-featured version.

Advantages of surveys:

- They can reach a large population.
- They can focus the users on specific topics.
- They can be used to prioritize requirements.

Disadvantages of surveys:

- It can be difficult to probe into negative responses.
- It may be difficult to get the surveys returned.

Best practices:

- KISS (Keep It Simple Stupid); few questions, easy to read.
- Encourage participation (through management support).

5.10 Discovery/JAD/Facilitated Sessions

This section covers facilitated sessions, discovery sessions, or JAD sessions, which are all different terms for basically the same thing. Simply put, it is a session where the stakeholders are brought together to reach consensus on what the system requirements should be.

5.10.1 History

JAD, which was originally called Joint Application Design, was pioneered by IBM in the late 1970s and developed into a formal development tool in the early 1980s. It was a part of the concepts of Rapid Application Development (RAD) as described by James Martin in the book with the same name, and it was also integrated with a lot of the computer-aided systems engineering (CASE) tools that were popular at that time.

Originally, there was Joint Requirements Planning as well as the Joint Application Design; however, over time, they have merged under the umbrella of Joint Application Development. It has also moved from being fairly specialized and integrated with design tools to a method that can be used in any situation where there are multiple stakeholders and where there is a need to reach consensus among them.

5.10.2 Characteristics of a JAD Project

Not every project is suitable for JAD. It makes sense only if there are many stakeholders and there is some potential for disagreement. It also requires that there is stakeholder buy-in to the process. JAD sessions can be high risk because they are based on confronting different opinions and creating a consensus solution. Emotions often run high, and if the participants have not bought into being at the session, it can be very difficult to keep it on track.

JAD sessions can be high risk because they are based on confronting different opinions and creating a consensus solution.

There must also be a willingness by the sponsor to delegate authority to the JAD team. If the solution has already been decided or if the JAD team's recommendations are likely to be overturned, then a JAD session may not be the way to go. JAD sessions are very time consuming, and if the results are ignored, it will be even more difficult to hold other JAD sessions in the future.

Finally, there need to be realistic expectations of the commitment necessary for a successful JAD session. It takes significant planning, training, and dedication of resources to do a successful JAD. If there is not enough time to do the JAD right, it is better not to do it at all.

5.10.3 What Is Created in a JAD Session?

The main goal of any requirements-gathering session should be capturing, documenting, and modeling requirements. JAD is a technique used to do that in a multi-stakeholder environment. While JAD in itself is a technique, it is also a combination of many other approaches, techniques, and tools. There is no one

approach or tool that is most suitable for a JAD session. The documentation tools may be activity diagrams, use cases, or data models, as discussed in [Chapter 8](#). There could be elements of idea generation (or brainstorming) as well as prototyping within the JAD session (both techniques are discussed later in this chapter). Like all of the techniques discussed throughout this book, JAD should be viewed as a tool for the analyst's toolbox that can be used in combination with other tools. Each situation will be unique, and any approach to requirements gathering should be customizable.

5.10.4 JAD Participants

Because JAD is based on group dynamics and consensus building, it is easy to see that the people participating will be critical to the success of the session. There are many potential participants in a JAD session; however, an ideal number is five to ten participants. Sometimes, the session may have fifteen to twenty people; however, with a larger group, it becomes questionable if it is really still a JAD session where everyone is heard. Some common roles for participants are described below.

5.10.4.1 Facilitator While the customer is the most important person when gathering requirements, the most important person to the success of the JAD session is the facilitator. A good facilitator can sense how well the team is functioning, guide the team when it is struggling, and ensure participation by all the team members. A poor facilitator will create chaos in the session and quickly lose control. It is one of those many thankless jobs where a good facilitator is not noticed but a bad facilitator becomes the center of attention for the session.

While the customer is the most important person when gathering requirements, the most important person to the success of the JAD session is the facilitator.

The facilitator should be neutral to the outcome of the project. In other words, the facilitator should have no interest in the result, only in the successful completion of the JAD session. If the facilitator is viewed as being partial to one customer group over another or favors the development organization, the facilitator's job will be made much more difficult. Facilitators can be brought in from a consulting organization—for large, high-visibility projects, that is a recommended approach. But for smaller projects, outside consultants are often cost prohibitive. An alternative often used is to get a person with good facilitation skills from a different project within the organization. As a last resort, one of the project team members (or even the project manager) may be used; however, make sure that the person chosen is aware that he or she is now playing a different role on the project. Instead of focusing on what the impact of the JAD session will be on his or her workload, the focus should be on getting the best possible outcome.

The following are some of the key roles the facilitator will play in the session:

- Timekeeper: The JAD session may last several days, and normally there are many agenda items to get through. The facilitator must be able to stick to the agenda, determine when it is time to move on, and decide which discussions need to be set aside for a later time. Consider having someone on the team, maybe the scribe or the business analyst, help with this by letting the facilitator know when the time is up for a topic.
- Mediator: Although conflict is a normal and healthy part of a JAD session, there is always a risk that it will move from healthy to unhealthy. It is the facilitator's role to sense when it is time to step in and mediate a dispute.
- Coordinator: There are many groups involved with the JAD session, and the facilitator needs to spend a significant amount of time scheduling participants, facilities, and other resources. It is more and more common that some participants may be attending via conference calls, videoconferencing, or the Internet. This adds another level of risk that must be carefully coordinated. If at all possible, it is recommended that all the participants attend in person.
- Counselor: This may be the most difficult part of the facilitator's job. When putting a group of people together in a room, some participants may be uncomfortable with the process, with the outcome, or with other participants. The facilitator, being in charge of creating a good environment for the session, may need to work with some participants on how to state their opinions and help them become part of the team dynamic.
- Salesperson (for the process): There will likely be naysayers and unwilling participants in a JAD session. The facilitator must be able to get buy-in and commitment to the process and to the techniques used in the session. If there is upfront agreement among the whole JAD team, it becomes the team's session and the team's deliverables rather than the developer's or the business analyst's.
- Clarifier: Not every participant will be a master communicator. Part of the role of the facilitator is to probe and to restate and check statements against other participants' understandings to ensure clarity in all requirements.
- Summarizer: After a lengthy discussion and before moving on to the next point, there should be some time set aside to recap and summarize what was discussed and what was agreed upon. That will minimize confusion down the road.
- Diplomat: The JAD session will include likeable people and probably some who are less likeable. The facilitator needs to be the rock that is stable in the middle of emotions and disagreements. The only way to be an effective diplomat is to be neutral in regard to the outcome. The focus must be on the parties agreeing, not on the facilitator steering or driving toward a certain result.
- Team builder: The facilitator must create an atmosphere that is professional and objective, but also where creativity can thrive and the team can work well together.

It sounds like finding a good facilitator can be a very difficult task, and that is true. That is why it is recommended, whenever possible, to use a professional facilitator. Some characteristics to look for when trying to find an internal facilitator include

- Strong listening skills
- Good speaker
- Able to read people (both speech and body language)
- Approachable
- Quick on his or her feet
- Calm demeanor

If you identify such a person in your organization, train and hold onto him or her as that person will be invaluable for the organization.

5.10.4.2 Business Analyst The role of the business analyst has already been discussed in general terms; however, in the context of a JAD session, his or her role will be to ask the right questions, ensure that the right stakeholders are in attendance, and provide input into the analyst's areas of expertise. The business analyst should work closely with the facilitator. It is actually common that the business analyst is the facilitator, but it is not recommended. The business analyst would not truly be neutral in regard to the result of the JAD session because that result is likely to have an impact on the analyst's future deliverables.

Going into the session, the business analyst needs to explain to the facilitator what the goals of the session are and agree on the best approach to reach those goals. It is typically the business analyst who prepares the questions for the session and decides what modeling and documentation will be used. During the session, the business analyst may advise the facilitator in regard to participants who need to provide more information and areas that may need more attention.

5.10.4.3 Scribe If the facilitator is the key to a successful JAD session, the scribe is the key to making sure that everything of value from the session is accurately documented. If at the end of a 3-day JAD session, there is a blank piece of paper for meeting minutes (or a very short document), it probably means that there is a reliance on memory that, even in the best of cases, is unreliable. It is recommended that there be two scribes in most JAD sessions. One of them should focus on the decision-making process and the conversation, and the other should document the work being done on flip charts and white boards. They can also tag team on those tasks, which makes it easier to pay attention.

It takes a special person to scribe well. The scribe needs to understand the topics discussed and to sort out what is important and what is not. It is not a speaking role in the JAD session, but the scribe can and should ask for clarifications when needed. Take time during the session, at natural transition points, to review the notes taken thus far. It is best to get consensus on what has been decided on while the areas of

discussion are still fresh in everyone's mind. If possible, bring the notes up on a screen that everyone can see and review point by point.

When selecting the scribe, look for the following traits:

- Good note taker
- Good business understanding
- Good understanding of modeling techniques used
- Ability to stay focused

5.10.4.4 User (Customer) With all the discussion about who is the most important person in the JAD session, suffice it to say that without the user, the need for the session goes away. Too often the session is attended by the people who are available rather than the people who are the most knowledgeable. Getting the right user into the session is not only important for the session itself, it also ensures that the result of the session will be accepted in the end. If the users who attend the session do not have the respect of the people in the organization, those people are not likely to accept the outcome.

Getting the right user into the session is not only important for the session itself, it also ensures that the result of the session will be accepted in the end.

Negotiate for the right user. There is a tendency to just accept the person selected by management, and it may be true that there is no option. But start by identifying the characteristics that would make the ideal participants, including criteria such as subject-matter knowledge, experience in the organization, reputation, and ability to work in a group environment. If those criteria have been communicated ahead of time, it will also be easier to come back and ask for a replacement candidate if the first one does not work out.

It is also important to have a cross-section of users. The experienced user can provide big-picture views and will know what is important to the organization. However, the novice user may be better at explaining what the difficulties are in today's environment. The things the novice struggles with on a daily basis may never come up when you talk to the experienced user who has learned how to work around those areas.

The user must be trained on all the techniques used in the JAD session and must understand the JAD process itself, as well as have a clear understanding of how to read and use any of the modeling techniques used in the session.

5.10.4.5 Subject-Matter Experts Sometimes, there will be a need to bring in SMEs to a JAD session. They may bring expertise on a topic of discussion, the capabilities of a new technology, or information about government regulations. SMEs are in the session to provide information, not to be decision makers. They typically stay for only a portion of the session.

When involving SMEs, make sure that they are acceptable to all parties in the session. If SMEs are brought in by one user to support his or her viewpoint, it must be ensured that the SMEs are objective and their only role is to explain facts.

5.10.4.6 Developer It is important for the developers to be at the JAD session to gain an understanding of the customer's thought process and to understand what will be asked of them. The developers, though, should not play a major role in the early JAD sessions. They must be given clear instructions on what to discuss and what to stay out of. In the early JAD sessions, the purpose is to understand what the requirements are, not how they will be implemented. Naturally, the developers' minds will be thinking about how this will be implemented, but that topic is for a different time, quite possibly a later JAD session.

In the early JAD sessions, the purpose is to understand what the requirements are, not how they will be implemented.

Part of a requirements JAD session is often to discuss potential solutions to make the discussion of systems less abstract. There may be a prototype or a similar product being used to frame the discussion around requirements. The developers will play a key role in developing and presenting those, but always keep in mind that the focus should be on *what* before *how*.

Moreover, be aware of the ratio between developers and customers. Some JAD sessions will have two or three customers and eight to ten developers. That is not a good ratio as the customers are likely to feel that it is a developer session rather than a customer session.

5.10.4.7 Sponsor The role of the sponsor varies from organization to organization. Here, *sponsor* is defined as the champion of the project within the customer organization. This may in fact be the paying customer, but it does not have to be. The sponsor is an escalation point for the project, meaning the person who removes obstacles on the customer side.

There are strong disagreements in regard to the sponsor attending the JAD session. Some organizations prefer to have the sponsor attend the kickoff session and then leave; in other organizations, the sponsor stays the whole time. There are pluses and minuses to both. On the plus side, with the sponsor there, you have a quick escalation process, and it shows the commitment of the organization. On the minus side, if the sponsor will intimidate other participants and stifle a free exchange of ideas, it is probably better to keep him or her out of the session.

Regardless of whether the sponsor attends the session, he or she is key to the JAD preparation process. It is much easier to get organizational commitment to attend JAD sessions and to send the best representative when the sponsor is asking for the resources. This is especially true if the sponsor is well respected within the organization. The sponsor should also be the one setting the stage in the kickoff meeting, laying out the goals and objectives and showing the organizational commitment to the effort.

5.10.4.8 Observers Observers are not really participants of the session, but are typically there either to learn about the project or to learn about how to conduct a JAD session. They can be future facilitators, business analysts, or scribes who are being trained in the details of how to conduct a JAD session. It is also possible to treat the developers as observers, especially in the early session.

The rules for the observers are simple: stay quiet and do not interfere with the session. The facilitator may ask for feedback during breaks; however, during the session, it will just add confusion if the observers are interacting with the participants.

5.10.5 JAD Process

Although there are many JAD processes defined in different books, a simple four-step approach as seen in [Figure 5.4](#) will be used as an example.

5.10.5.1 Establish Goals and Objectives The JAD session is a subproject within the analysis phase and needs to be treated as such. There can be one or many JAD sessions conducted to capture requirements. If there are multiple sessions, there should be goals and objectives established both for the overall effort and for each individual session. Do not just state the goal as *gather requirements* for multiple sessions. Rather, each session should have a unique objective and a unique deliverable. Otherwise, there will be a tendency by the users not to get serious until the last session.

5.10.5.2 Prepare for the Session The JAD checklist in [Figure 5.5](#) is a tool that can be used as a starting point for planning and executing a JAD session to help ensure nothing is forgotten. Because every JAD session will be unique, make sure to customize the process to fit the specific environment of the current project.

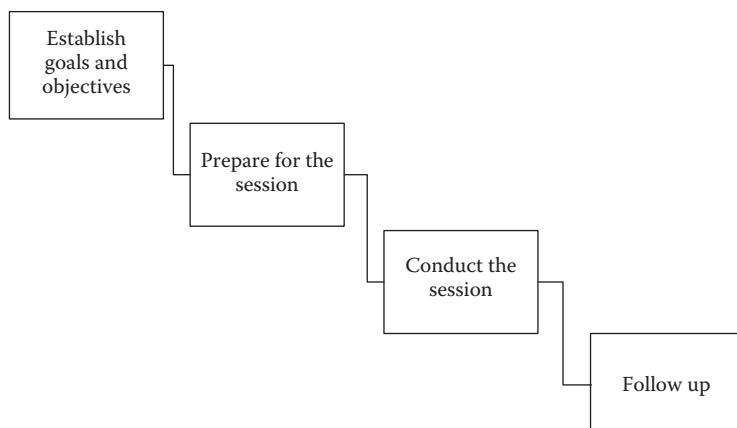


Figure 5.4 JAD process.

Initial

- Define who the sponsor is.
- Determine the goals and objectives.
- Establish participating organization.
- Determine what process to use.
- Determine who the project leader is.

Research

- Become familiar with the business environment.
- Research similar products.
- Select participants.

Workshop plan

- Hold kick-off meeting.
- Create agenda.
- Prepare materials.
- Prepare room.
- Create straw man (initial thoughts to kick off session).
- Train participants.
- Script the session.
- Dry run of session (practice).

The session

- Opening speech.
- Review agenda.
- Review objectives.
- Establish ground rules.
- Get buy-in to process.
- Conduct the session.
- Document issues.

Follow up

- Complete documentation.
- Present result to sponsor.
- Evaluate the process.
- Follow up on issues.
- Feedback of end result to participants.

Figure 5.5 JAD checklist.

Select a location where interruptions can be minimized. It should be away from the customer work environment. There should be enough room to comfortably facilitate the session, and it is a plus if refreshments can be provided. Well-fed people tend to be more open to consensus building. A sample room layout is shown in [Figure 5.6](#).

Notice the U-shaped layout of the tables. This serves three purposes. First, it allows everyone to see everyone else. Second, there is no head of the table, so everyone is equal (at least from a seating standpoint). Third, it gives the facilitator great access to all participants, especially if there are any problem participants. If there are two people having a side conversation, it is easy for the facilitator to walk into the U and

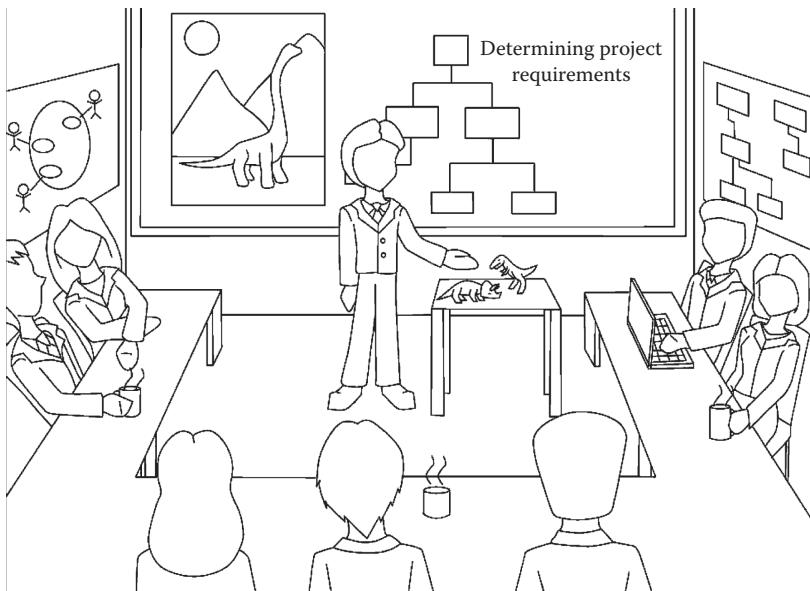


Figure 5.6 JAD room.

stand right next to the people talking while continuing the facilitation. Most people (but not all) will take the hint and quiet down.

Decide and communicate what the format of the deliverable from the session will be. Will it be a text document? Is there a template being used? What modeling techniques will be used in the session? The participants must know this going into the session, and in some cases they may need training to become familiar with some of those deliverables and modeling techniques.

Create an agenda for the session and send it out to the participants ahead of time. Make sure to include some contingency time in the agenda. Have some extra time added to the end of the day's activities, or try to place some activities on the agenda that can be skipped if there is a need for it. Do not make the agenda fluffy, because then the participants will not take it seriously. Plan for surprises. Allowing the participants a chance to review the agenda ahead of time gives them a chance to voice any concerns in regard to why they should attend the sessions, and maybe even suggest some other people who should be attending. Those are concerns that are better addressed before the session rather than during it.

5.10.5.3 Conduct the Session The session can be started with a kickoff meeting, if that has not already taken place. It is recommended that the kickoff is made a separate event; however, that is not always possible owing to time restrictions or sometimes because of people traveling in for the JAD session. There should be some form of ice-breaker in the beginning of the session. Sometimes, this is viewed as too juvenile and is skipped, but there is a real need to mark this as the beginning of a new endeavor for the team and as an occasion to establish team identity.

Also make sure to establish ground rules up front. There are some obvious ones, such as

- Turn off (or silence) cell phones.
- No personal attacks.
- Return on time after breaks.

But there are also some ground rules that will vary from session to session. In some sessions, the team may limit discussion on any one item to 10 minutes; in others, the team may decide to go until consensus has been reached. Also, in some sessions, the team may vote on a decision; in others, there may have to be full agreement by the whole team. When establishing the ground rules, they should be suggested and approved by the team itself. The facilitator may suggest one or two to get the team started, but the real decision and buy-in must come from the team. That way, the rules belong to the team and are more likely to be enforced by the team, making the facilitator's job much easier.

When conducting a JAD session or any team activity, it is good to be aware of what the state of the team is. Are the members working well together? Are they fighting? Are they even talking? A good model for the stages that a team goes through is shown in [Figure 5.7](#). The figure shows the how the team functions at different stages, from the first time the members meet each other until they are a fully functioning, productive team. In the forming stage, the team members are hesitant, there is limited interaction, and no one wants to speak up for fear of looking stupid. The facilitator needs to do more controlling at this stage. There should be specific activities, icebreakers, and team assignments to force the team members to open up and start interacting. Once they do, they move into the storming stage. This stage can be uncomfortable for the team, but it is critical. Here, the team members start testing the knowledge and resolve of other team members. Arguments are frequent, and tempers often flare. The role of the facilitator is to minimize bloodshed, but also to step back and let the team sort it out. As the issues are sorted out, the team moves into the norming stage. It is during this stage that the members of the team start falling into their natural roles and responsibilities, and utilizing the strengths of the team. Once that is complete, you have a performing team. At this point, the facilitator's job becomes easier and the team pretty much manages itself, with the facilitator just guiding. Be aware that as soon as the topics of discussion change or the team members are changed, the process

<i>Stage</i>	<i>Group functions</i>	<i>Leadership style</i>
Forming	Poorly	Autocratic
Storming	Poor to OK	Guiding
Norming	OK	Democratic
Performing	Well	Hands-off

Figure 5.7 Forming/storming/norming/performing.

starts over again. Good facilitators learn to observe the team and determine what stage it is in, and then adjust their facilitation style on the basis of that.

Once the session is underway, the facilitator's job is to assess whether the goals and objectives of the session are being met. Those goals and objectives should be posted in the JAD room and used to determine if a discussion pertains to meeting those objectives. There will be many topics that are near and dear to the participants; they may even be important for the project itself. But at this point, if they are not relevant to the objectives of this session, then they should be set aside. A tool that is useful for these situations is the *parking lot*. The parking lot should be on a white board or flip chart, visible to all in the session. Whenever a topic is brought out that is not part of the scope of this session or when a topic has been going on for too long and it needs escalation, then it is put in the parking lot for later action. Although it is true that in many cases, there is a plan to come back and look at the items in the parking lot, quite often it is just a technique for the facilitator to get the group to move on and accomplish the actual objectives of the session. It is sometimes easier to diffuse an issue and move off a topic by telling a strong-minded stakeholder: "Yes, that's a valid point, but outside of the scope of this session. Let's put it in the parking lot and assign someone to look into it after the session."

Time keeping is another focus area for the facilitator during the session itself. It often works well to delegate this to someone on the team who will track the agenda and the clock, and point out when it is time to wrap up a topic. So, what to do when it is time to end a discussion and the team wants to keep going? This is actually a common problem, and there are different options to consider depending on the circumstances. Some of the more popular are as follows:

- Assign a subteam of the JAD group to take the discussion offline and come back with a recommendation. This is often a great way to solve the issue unless the other topics in the JAD session depend on a resolution of this issue.
- Review the agenda for other items that can be deferred while the current discussion is extended. There should be some contingency set aside on the agenda by the facilitator for this type of occasion. There may also be a topic later on the agenda that can be shortened or deferred to a subteam.
- End the discussion and vote on a decision. Although this is not usually the best way to go, sometimes enough is enough, and it is time to make a decision and move on. The decision itself should not be the facilitator's; however, a strong facilitator can drive the team to make a decision on the issue being discussed.

One of the main purposes of a JAD session is to build consensus. Having consensus obviously makes the rest of the project much easier and improves the odds of having strong buy-in to the final solution. But how is this actually achieved? First of all, there must be a clear definition of what is meant by consensus. Does it mean that everyone on the team will love the decision and agree that it is the ultimate way to go? Or does it mean that the participants agree not to agree and vote on the outcome? Or is it that

even though it may not be what each participant would have preferred, they all agree to support and live with the decision outside of the JAD room? Although it is up to the team to decide, and voting is a common option to make a decision, the recommendation for a real consensus is that each person is willing to support the decision outside of the room.

In my past experience working with Saturn Corporation, I saw a great example of having a corporate focus on operating through consensus. In their JAD sessions, they would often use a red/yellow/green approach to reach agreement. It is an approach fairly common in the industry. The basic premise is that when a decision is needed, each team member has to state where he or she stands and why. The three options are

1. Red—against the decision and not willing to support it. However, it is not enough to just show red. It is also the responsibility of the person showing red to let the team know what it would take to get him or her to yellow.
2. Yellow—not in favor of the decision but willing to support it. Again, this person needs to state what it will take for him or her to get to green.
3. Green—in favor of the decision.

It takes patience and trust to build consensus. It is likely that in the first few JAD sessions, there will be hesitation and some game playing by the team members to try to get it their way. There is a tendency in most organizations to have a win–lose attitude, meaning that for one party to win, the other party must lose. For the consensus process to work, that attitude must change to one of win–win. There is a need to listen, open mindedly, to the opposing views and to find a way to tweak the decision, or to augment it, to find an option that both parties view as a good option.

There is a tendency in most organizations to have a win–lose attitude, meaning that for one party to win, the other party must lose. For the consensus process to work, that attitude must change to one of win–win.

On the Prescription Tracking Project, Ray, the pharmacy manager, wants to be able to track how long it takes for each pharmacist to fill a prescription. Meanwhile, Diane, the pharmacist representative, is uncomfortable with that; she feels that it will encourage the managers to view it as a way to measure individual performance and fears that it may lead to the pharmacist sacrificing customer satisfaction to rush the fulfillment of the prescription. After further discussion, it is found that Ray really does not care about evaluating an individual's performance, but rather wants to evaluate the overall efficiency of the pharmacy. Diane does not mind the measurement, but does not want it attached to a certain individual. After further discussion, it is decided to attach the time measurement to the prescription drug rather than to the pharmacist. That satisfies both parties' goals. Although it is not realistic to always find a solution that satisfies everyone, it is possible more often than not. It takes a willingness to listen and to be flexible with expectations.

One final point on conducting the JAD session is that because it is often lengthy, there will be a need for breaks and maybe for lunch as well. Sometimes, it is difficult to get the team back to the room again after breaks. Make sure that one of the ground rules deals with breaks and sets the expectation to return from them on time. If there is a need for the JAD team members to check back in with the office during breaks, it may be smarter to have less frequent but longer breaks, allowing for checking e-mails and voice mails.

5.10.5.4 Follow Up There will be a lot of documentation coming out of a JAD session, especially a session that may have gone on for 3 days or more. Do not try to send out a play-by-play review of the session, rather focus on the decisions made and the rationale for the decisions. If the scribe did a good job, and if there were frequent reviews of the documentation during the session, this final deliverable should be ready to go very shortly after the JAD session. Sometimes, there is a desire to wordsmith the documentation to make it perfect. Although it certainly should look professional, time is of the essence and a normal JAD deliverable should be ready to go out to the team within a day of the session. This excludes any prototype development needed if that was a part of the deliverables. The first round of the follow-up should go to the JAD team members only, with a request for a quick review and approval. Once that has been accomplished, the document can go out to a wider distribution. Then, if there are disagreements, the JAD team is formally behind the document. Some of the outputs from the JAD session, such as models, prototypes, and decisions made, should also be added into the BRD.

5.10.6 Facilitation

The importance of strong facilitation for a successful JAD session has already been stated. Although it is recommended that a professional facilitator be used, most projects will need to find and train an internal resource. In my JAD workshops, there is an activity on facilitation fears and another one on problem participants. Over the years, those activities have generated a wealth of information that demonstrates that most people have some fears about being a facilitator, and also that most of us have been exposed to problem participants many times in both JAD sessions and regular meetings.

The most common facilitation fears are

- Fear of public speaking
- Fear of losing control of the session
- Fear of not knowing the topic
- Fear of silence
- Fear of not meeting objectives

What can be done to deal with these fears? The simplest answer is to prepare. Because JAD sessions are high risk and high pressure, the facilitator must set aside

sufficient time to prepare for the session. A facilitator must practice facilitation to be successful. The first time a facilitator is doing a JAD session with a customer should not be the first time he or she is facilitating. Practice by facilitating meetings and doing dry runs (practice runs) of an actual JAD session with the internal project team. This will give practice both for facilitation techniques and the modeling and documentation approach that is planned for the session. Another way is to join public speaking organizations where the facilitator can get used to talking in front of a group.

Meet with all participants ahead of time. If the fear is losing control or a silent team, knowing the participants and having met with them ahead of time will help minimize this. It is important for the facilitator to have a good understanding of the topic of the session. It is not good if the facilitator looks lost during the session. However, the facilitator is not intended to be the expert, but must have enough knowledge to be able to assess how well the session is going. Beyond that, a good facilitator will make use of the knowledge of the team.

Make sure that each of the JAD participants is clear on what the objectives are. It is also worth remembering that not all objectives are reachable. The purpose of the session may be to gain consensus on requirements, but there may be mutually exclusive interests among the participants, in which case there may be a need to escalate to the sponsor.

The term *problem participant* is probably better avoided, even though that is what it often feels like when the JAD team appears out of control in the middle of the session. It is better to focus on problem behavior. One reason is that behavior is easier to change (sometimes) than the actual participants. What are some common behavior issues and what can the facilitator do to deal with them?

- The talker: Some people talk more than others. It is never the intention to have equal participation in a JAD session, but it is the intention to have equal opportunity for participation. If someone monopolizes the discussion, the facilitator must address it. Within the session, the facilitator can steer the conversation to other participants by asking them to comment on what has been said. Outside of the session, on a break, take the talker aside and tell him that although you appreciate his ideas and enthusiasm, you also need to allow some of the less outgoing people to be comfortable voicing their comments. If this is done correctly, the talker can become an ally of the facilitator.
- The quiet one: Some people are quiet because they are uncomfortable speaking up and some are quiet because they do not have anything to say (a rare but desirable quality). Be careful about putting too much spotlight on this person as that may just cause him or her to withdraw even more. This is a case where it helps if the facilitator has actually met with the person ahead of time and gotten to know his or her personality. Try to find a topic that this person has shown an interest in during conversation before the session, and use that to

draw the person in during the session. A good icebreaker in the beginning of the JAD session can also help with this.

- The disagreeer: This is not a real word, but maybe it should be. The disagreeer is easy to find in a meeting. Although there can be many causes for disagreement, the one that is important here is opposition to the project. The disagreeer does not like the project, does not agree with the objectives, and would be very happy if the project were cancelled. Start by doing some root-cause analysis. What is the reason for the disagreement? Does the person feel there is a better way, or that he or she should have been involved earlier? Or is this a project that will, if successful, have some negative impact on his or her working conditions?
- The one nobody understands: Some people have difficulty articulating their ideas. This is probably the most sensitive situation to deal with. It takes a lot of patience from the facilitator and some hard work. Try restating what the person said and ask for confirmation. Prototyping and using live examples may also help narrow down the discussion. The goal is to understand the point the person is making without making him or her feel uncomfortable.
- Side bars: A JAD session can quickly deteriorate if there are multiple conversations going on at the same time. That said, it is normal in a creative environment to have some one-on-one interaction between the team members. The facilitator must find the balance between too much control and too much chaos.
- The attacker: This is the person responding to someone else in the session with “That was the stupidest thing I heard you say since the last time you opened your mouth.” When this happens, the facilitator must step in. It is easy for a session like this to deteriorate into name calling, and this person would definitely move it in that direction. First of all, there should be a ground rule dealing with personal attacks; if there is not, it is OK to add it at any point. It is also recommended that the facilitator raises the issue with the person during a break. If it is a serious attack, it may be worth taking the break right then to make sure that the session does not get off track. Although it is OK to address it in the session itself, be careful with that because it may just make the person defensive. One-on-one communication tends to work better for negative conversations.

5.10.7 Summary

Few people are successful with their first JAD session. But after a toolset is developed that the participants are comfortable with and after the organization gets more comfortable with the concepts, it can be a major improvement to an organization's requirements gathering. It does take a large investment in training and in time, so

JAD should not be something that is introduced to an organization unless there is a strong commitment to it.

Advantages of JAD:

- There is user involvement and ownership.
- All players hear the same message.
- Disagreements are solved at the session.

Disadvantages of JAD:

- It can generate conflict.
- It is difficult to facilitate.
- It takes a high level of organizational commitment.

Best practices:

- Select a top-notch facilitator.
- Use more than one scribe.
- Hold the session offsite.
- Have clear goals and objectives for each session.

5.11 Focus Groups

Focus groups are similar to JAD sessions, but the group in the session is advising, not decision making. Focus groups are typically used when developing a product for a large user base to find out what features most users are interested in. There can be more people in a focus group than in a JAD session because there tends to be less controversy. The outcome is not a consensus decision, but rather some ideas for the team to consider. The sessions tend to be shorter, and they can be held with people from one area of the population (all accounts payable clerks or all executives), which is called a homogeneous focus group, or they can be with a cross-section of people, called a heterogeneous focus group.

As with JAD sessions, the focus groups need to be well facilitated, but it requires a different form of facilitation. Because the purpose is to gather opinions and ideas, it is important to create a positive atmosphere in the room. There is not as much concern about participation, and if someone wants to mentally check out, that is not a disaster either. With focus groups, it is more common to look for volunteers, trying to find the people who are interested in enhancing the product on which they are working. Some of the ideas coming out of a focus group can be used as input into later decision-making JAD sessions. The players in focus groups are pretty much the same as in a JAD session except there is a larger group of users.

When using homogeneous focus groups, the session typically deals with the details of a process or work function. A detailed flowchart or process model may be a good modeling technique to use in this type of session to stay on track. For a heterogeneous

focus group, a workflow or swim-lane diagram may work best because the main interest here is on communication or interfaces between functions and organizations. These techniques will be explored in [Chapter 8](#).

Advantages of focus groups:

- They generate a lot of ideas in a short time frame.
- They can elicit input from a large number of existing users.

Disadvantages of focus groups:

- They can deteriorate because participants are not decision makers.
- They are difficult to keep on track.

Best practices:

- Set realistic expectations up front. The participants must realize that their recommendations may not make it into the product.
- Keep sessions short (2–4 hours).

5.12 Market Research

It is common that organizations will take the BRD and use it to help evaluate if they want to build a system in-house, outsource the development, or buy a package. It is often overlooked, though, that those same packages can be used to extract requirements. Especially in the early stages of requirements gathering, it can be very productive to review the features and functions that exist in commercial off-the-shelf software. It can serve as an initial prototype to make the systems discussion less abstract.

Before selecting one or more packages to evaluate, there are some important recommendations to keep in mind:

- Set the customer's expectations. This is not intended as a preview of what the customer will get. For a large company, the package may be too simplistic, and the customers must understand that this is just to kick off the requirements gathering. For a small company, this can also be an issue, and sometimes a more serious one. If the customer is shown functionality that would be too expensive and complex to implement, the eventual final product may be poorly received, because the customer has already seen something nicer. Imagine walking into an automobile dealership looking for a car and the salesperson brings out a Corvette for the test drive. By the time the actual deal is signed on a four-door mid-size, there is probably some disappointment on the part of the customer.
- It can be very time consuming to locate a package and to bring in a demo version (or a trial product). If this is a small project with a tight timeline, this may

not be a realistic option. There also must be some subject-matter expertise on the product, often provided by a consultant; but again, there is a cost associated with that. If there is a package that may fit and that will be considered, then there is an option to try to get the vendor involved with the evaluation. However, remember that the vendor's objectives and the project's objectives are usually different.

- It can limit creativity. By looking at an existing product, the customer will get some preconceived notion of what the product should look like. It may limit the thought process, it being easier to just go with what the package has versus determining what would be the best solution for this organization.

Advantages of market research:

- It provides a sample of a working solution.
- It can be used as a prototype.
- It finds *best practices*.

Disadvantages of market research:

- It limits creativity.
- It can be time consuming.
- It may have additional cost.

Best practices:

- Set the level of the customer's expectations.
- Select a package with a target market matching the customer's organization.
- Be aware of vendor help.

5.13 Evaluate *Best Practices*

Best practices can involve market research such as discussed previously, but it can also involve reviewing what the competition is doing. If the project is an accounts payable system, there is probably not much difference among organizations, but if the project is market analysis or data warehousing, then it may pay to look at what the leading companies in the market are doing. How does one determine what a *best practice* is? Look for the industry leaders in the same industry as the customer. Traditionally, in the automotive industry Toyota has been studied for best practices. This can be a chance to look outside of the immediate industry as well. If the project is to develop a new ordering system, maybe Amazon.com should be studied. If it is logistics, then maybe FedEx or Wal-Mart would be worth looking into. Make sure that the customer agrees with the evaluation of what the *best practice* organizations are.

Sometimes, a wealth of information can be gotten from trade journals and trade shows. A business analyst with broad industry exposure is key in this area. If that person does not currently exist within the organization, there may be a need to bring in a consultant.

Advantages of evaluating best practices:

- It is not reinventing the wheel.
- It allows making significant improvements in a short time frame rather than gradual enhancements based on current environment.

Disadvantages of evaluating best practices:

- Information may not be readily available.
- Information may be skewed depending on the source.

Best practices:

- Use trade shows and trade journals.
- Use consultants.
- Create business analysts within the organization with broad industry knowledge.
- Ask the customer “Who’s the best?”

5.14 Prototyping

Prototyping is one of the most popular tools used for systems development and requirements gathering. A stumbling block often encountered when discussing requirements with stakeholders is the abstract nature of systems. If the customer is buying a car, he or she can go for a test ride and decide which features he or she likes and which ones should be avoided. If the project is the construction of a house, the customer can meet the builder in a model home to visualize and decide on features. However, with a system, this is a difficult process. Prototyping can help with the visualization.

Prototyping can help with the visualization.

James Martin defines *prototyping* in his book *Rapid Application Development* as “a technique for building a quick and rough version of a desired system or parts of that system ...” that “serves as a communication vehicle.” This is an important definition, because the main purpose of prototyping is to enhance communication. The impact of those statements is often forgotten and the prototype becomes the product in itself. It is constantly refined and perfected with so much time spent on it that it becomes difficult to abandon, or even to criticize it. A question I always ask my students when discussing prototypes is, “What is the worst thing that can happen when customers see the prototype?” Typically, the answers are, “They hate it,” “They want to change it,” or “They think it is too basic.” None of these is the answer I am looking for. Actually, the worst thing that can happen is that they love it, they will take it as is, just implement it tomorrow and the project will be finished! This is an indication that the analyst has done a poor job managing the customer’s expectations. The customer must be told ahead of time that this is just a shell, held together by sticks and chewing gum,

and that there is no way that this will be implemented for quite a while. It is possible it may never be implemented, because it is a communications vehicle and may even contain features that are not technically feasible, or at least not in sync with the technical architecture of the organization.

It is worth noting that in some projects, the prototype will become the product. Some methodologies (described in [Chapter 4](#)), including RAD and Dynamic Systems Development Methodology, are based on evolutionary prototyping where eventually the prototype evolves into the product. The prototype may be *good enough* or at least close to it. That approach is OK, but that possibility of implementing the prototype should be decided on up front. If developers know that the prototype may end up in production, they can prepare for that by following standards and incorporating required interfaces, at least at a conceptual level. In many cases, implementing a prototype may lead to problems down the road, but it can work, especially if the prototype was developed with that in mind. Make sure to do a risk assessment when moving in that direction.

Start the prototyping effort by establishing realistic objectives. Determine what is expected as a result of developing the prototype. Is it to determine requirements for the user interface? Is the customer's workflow being evaluated and analyzed? Is it trying to decide which high-level features should be included? On the basis of the answers to those questions, the prototype may go in very different directions.

In the early days of prototyping, there were two main types of prototypes:

- Evolutionary, where the prototype gradually gets refined and eventually becomes the final product. This type requires more effort in creating the prototype, because the system being developed may eventually become the production system. However, it does have the advantage of not having to start over.
- Throwaway, where the prototype is not intended to be used for anything other than communications. The advantage is that it is very fast to develop. It can be done using just paper and pen or a white board, and allows requirements to be captured at a quick pace.

A different way of classifying prototypes is by looking at the scope. Is it a broad scope or a narrow scope? This can lead to the following two types of prototypes:

- Horizontal—shallow but broad. The purpose here is to make sure that all business functions are included, but there is typically not much working detail below the surface.
- Vertical—focusing on a small portion of the system, but going down into a lower level of detail. This may be done to try to understand how a function would flow at a detailed level, or it could be part of a feasibility study to figure out if a certain technology will work.

Although there are some sophisticated tools for prototyping, with more of them coming into play all the time, many prototypes developed today are using screen prints, frames, or PowerPoint®, which is really more of a storyboard approach (covered

in the next section). It is fast and simple, and for the most part, it is obvious to the customer that this is not the real system.

Advantages of prototyping:

- It is a good tool for communication with customer and developer.
- It involves the customer.
- It reduces abstractness.
- It can help determine feasibility of a solution.

Disadvantages of prototyping:

- It can result in unrealistic expectations.
- It can take a significant amount of time to develop, especially if it is an evolutionary prototype.
- It has a different look and feel than the final product.
- It may need additional resources from the development team, which may slow down progress or increase cost.
- It is easy to focus on *how* rather than *why*.

Best practices:

- Set realistic customer expectations.
- Define the purpose of the prototype up front.
- Do not over-engineer the prototype.

5.15 Storyboarding

Some analysts would view storyboards as a form of prototyping, and technically they are. The line between a storyboard and a prototype is vague. For the purposes of this discussion, a prototype uses a tool, has some level of functionality, and has a higher degree of sophistication than a storyboard. The primary purpose of a storyboard is to show what the user will do with the system. It can be created using a tool such as Visio®, or it can be done with white boards or Post-It™ notes.

Early in the requirements-gathering process, it is often better not to use a computer-based tool with screen prints and flows. In the early stages, the solution is still up in the air, and there is actually a good chance that some of the things being discussed may not even be automated. Using a sophisticated tool will often put customers in the frame of mind of “what they see is what they will get.” For the initial set of storyboards, use the following process:

1. Identify main scenarios within the scope of the project. These scenarios should outline the key functions for which someone will use the system. If use cases have been developed, then they can often be used for the initial scenario identification. (Use cases are covered in [Chapter 8](#).)

2. Determine which scenarios need to have a storyboard developed. Just because it is something that a user needs to do, does not mean that it should be storyboarded. Many scenarios are obvious and a storyboard does not add value. A storyboard should be specific: its purpose is to allow the requirements-gathering process to focus on something that is critical to the project. For the Prescription Interaction Project, some scenarios worth exploring may be
 - a. An existing customer wants to fill a prescription for a medicine that has a number of potentially harmful interactions, but none with any of the drugs from the patient's history.
 - b. The same as (a), but for a brand new customer.
 - c. The same as (a), but the drug has the potential to interact with a different prescription that has been used by this customer.
3. Meet with the stakeholders who will provide the requirements and outline the steps in the storyboard. Reach consensus on the high level of the flow.
4. Develop more detail on each part of the storyboard. Use a template to cover the main information that should be captured. Include elements such as
 - a. Storyboard identification
 - b. Description
 - c. User(s)
 - d. Trigger of the process
 - e. Inputs
 - f. Outputs
 - g. Issues
5. Validate with users and developers. Regardless of who was involved with defining and documenting the scenario, it must be reviewed with all key stakeholders. This includes the developer and the user, but also representatives from other functional areas that may be impacted by the scenario.

When developing storyboards, it is easy to get bogged down in details, exceptions, and error conditions. Sometimes, this will take an intuitive and user-friendly approach and turn it into an exercise in frustration. When working on the storyboards, always focus on the *happy path* initially. The *happy path* tells what will happen most of the time when the process is successfully executed. It ensures that there is a correct understanding of what the process should do before the focus becomes “what could go wrong.” It is OK to take some of the more complicated and involved alternative paths and errors, and storyboard them as well; however, it is good to be restrictive about when to do that. The customers and the developers often love a little bit of storyboarding, but as with most things, it can easily get overdone. Some signs of this are

- The customer loses focus and participates less.
- The storyboard appears to have a lot of duplication.
- The storyboard becomes the requirements document.

For the last point, it is important to remember that the storyboard is a tool to help with capturing requirements; it is a communication tool, often a very good one. But the storyboards are not the requirements. They do not necessarily even need to be a part of the BRD (even though it may help). In many cases the storyboard may be used again when the test cases are developed.

Advantages of storyboarding:

- It can be used early on to reduce abstractness.
- It is intuitive for the customer.
- It can be done without tools.

Disadvantages of storyboarding:

- It has a different look and feel than the final product.
- It is easy to focus on *how* rather than *why*.

Best practices:

- Keep it simple, especially in the early iterations.
- Keep customers involved.

5.16 Idea-Generating Techniques (Brainstorming)

Like prototyping, brainstorming and other idea-generating techniques can be used with virtually any requirements-gathering technique. They are often used in JAD sessions and focus groups as well as in traditional interviews. In this section, brainstorming and some of its common variations are explored. The variations to be covered here are

- Basic brainstorming
- Anonymous brainstorming
- Affinity diagramming
- Brainstorming with one person
- Brainstorming for a virtual team

5.16.1 Basic Brainstorming

This starts with a group of people in a room working with a specific problem, trying to generate solutions to the problem. The purpose of brainstorming is to use group synergy to generate as many ideas as the team can come up with, and then have the team build on those ideas to generate even better ideas. Sometimes, it is said that there are no bad ideas. That is obviously not true; there are a lot of really bad ideas out there. But allowing them to be raised and captured in the brainstorming session gives other participants an opportunity to tweak them and build on them to generate better ideas. The process to be followed is

1. Clearly state the problem. One of the difficulties with brainstorming is to keep focused and to stay within the boundaries of the problem that you are trying to solve. Make sure that the problem statement is clear and that all the participants agree with it.
2. Define the ground rules for the session. The ground rules should typically include “no disagreements allowed,” “all ideas are captured and visible,” and “allow everyone to participate.”
3. Start the brainstorming. The facilitator’s role is to make sure that each idea is captured and is stated in a way that is agreed to by the person stating the idea, and that is understood by everyone. Be careful about the difference between clarifying an idea, which is valid, and questioning an idea, which is not allowed at this stage.
4. Run the session for a predetermined amount of time, or until the team is running dry on ideas.
5. Go through the list of ideas and combine any duplicates. Also evaluate and see if there are any ideas that everyone agrees should be removed. This could be because they are outside the scope of the brainstorming session, they are better included within a different idea, or after further thought were realized not to be very good. Make sure, though, that everyone agrees that the idea should be eliminated.
6. Identify conflicting ideas and make a decision. When doing the brainstorming, it is likely that there will be two or more ideas that are conflicting. Go through the consensus-building approach discussed above for JAD sessions and make a decision on how to handle this. If the team is not empowered to make the decision, then document the ideas as conflicting and pass that information on to the decision makers.
7. Organize the ideas into a format that is easy to read. If there are only a small number of ideas, this may be very quick; however, if there are hundreds of ideas, this needs its own process. One approach is the affinity diagram, which will be covered later in this section.

The purpose of brainstorming is to use group synergy to generate as many ideas as the team can come up with, and then have the team build on those ideas to generate even better ideas.

When brainstorming, facilitation is important, both for what the facilitator does (staying focused, encouraging participation) and for what he or she does not do (question ideas, laughing at ignorance). During the session, the facilitator must watch out for *group think*. In most groups, there are some people who want to take control and some who sit back and wait. When the session is hijacked by a few participants, they tend to take the whole group with them in a certain direction. This can stifle creativity and a lot of potentially great ideas may be overlooked. One way to address this problem is through anonymous brainstorming.

5.16.2 *Anonymous Brainstorming*

One rule of brainstorming is for all to leave their egos at the door. Although this is a good intention, it is not always easy to achieve. Anonymous brainstorming is a way to get the ideas captured without knowing who actually stated the idea. An idea submitted by the inexperienced intern may be ridiculed and ignored, while an idea by the sponsor is likely to be considered (regardless of its merit). There are different ways to make brainstorming anonymous. This section will present a low-cost and a high-cost alternative, but with some creativity there are many ways between the two alternatives to achieve the same purpose.

The low-cost approach is to use the analyst's favorite tool, the yellow Post-It note. At the beginning of the session, give everyone a set of Post-It notes and start the session by having everyone write down five ideas. Each idea should be on its own note. Then collect the notes and shuffle them before writing them down on the white board (or flip chart). Once all the ideas are displayed, give the participants a few minutes to review them and then ask them for two more ideas. Go through the same process again and keep going until participants run out of ideas. Some guidelines to ensure anonymity are

- When some participants do not have any more ideas, ask them to write down *no idea* on the note and still collect them. That way, there will be notes from all participants.
- Do not ask for clarification of ideas. Rather, if an idea is not understood, ask the participants to rephrase it for the next round.

The high-cost approach is to use what is known as a *creativity lab*. There are a number of them around the country, run by different organizations. IBM has three or four of them. These are facilities that can be rented for the duration of the session. They come fully equipped with facilitators experienced in brainstorming and a room setup where each participant has a computer to use. The ideas are then typed into the computer, displayed on a screen at the front of the room, and seen by everyone. This allows more of a regular brainstorming environment where, when an idea is shown, the rest of the room can react to it and build on it right away. However, there are some disadvantages as well. If one person is continuously typing and the rest of the team is not, it may reveal whose idea it is. Typing errors can also be revealing here. One good compromise is to send the ideas through a moderator who checks spelling and displays the ideas at even intervals.

5.16.3 *Affinity Diagramming*

The purpose of the affinity diagram is to group ideas together into categories. Affinity diagrams can be created bottom-up or top-down. For the bottom-up approach, start with all the ideas displayed to the team, then let the team start moving the ideas around,

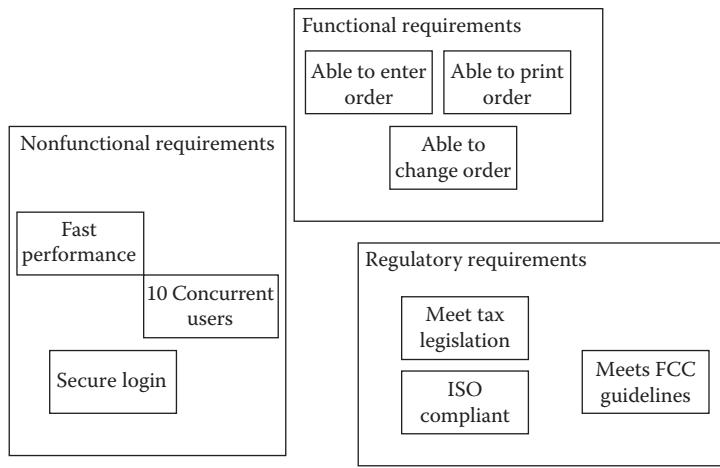


Figure 5.8 Affinity diagram.

putting those that have similar topics next to each other. Some people recommend doing this in silence, to minimize the chance for the more vocal people to take control; however, in most situations, it works well for the team to be talking while doing this. It is likely that some ideas will be moved back and forth because they may relate to other ideas. When that happens, review the idea and see if it is really a composite idea that can be split. If it is not a composite idea, then pick one group and put the idea there. Do not put the same idea up on the board twice, as that will potentially lead to duplication of effort down the road. Once the team is satisfied with the groupings, create a category name for each group. The result is an affinity diagram, as seen in [Figure 5.8](#).

The top-down approach starts with identifying key categories. Those categories can be based on organizational units (purchasing, marketing, and manufacturing) or on product deliverables (performance, features, training, and support). The advantage with the top-down approach is that it is possible to reuse a structure that has been successfully used in the past, and it is easier to define categories that can be assigned to different teams for further evaluation. The disadvantage is that because it is a pre-defined set of categories, it may limit creativity and the team may assume that all ideas must fit within these categories. If the top-down approach is used, also create a category called *other* to help track ideas that are outside of the expected categories.

The affinity diagram will help with categorizing requirements and prioritizing requirements. It is often better to prioritize requirements in groups rather than prioritizing each individual requirement. For example, there may be hundreds of usability requirements and dozens of performance-related requirements. Prioritizing each requirement would be tedious and time consuming. It is often sufficient to prioritize groups of requirements.

5.16.4 Brainstorming with One Person

Because brainstorming is based on group synergy, performing it with an individual sounds like a contradiction. However, it is possible to simulate some of the benefits of brainstorming

by asking the person being interviewed the same question more than once. Document each answer on a board or flip chart, and repeat the question again. The thought here is that by digging deeper, the interviewee's thought process may be triggered by earlier ideas, allowing him or her to come up with more creative ideas and be more likely to think *outside the box*. Eventually, the ideas will start being silly, and that may be a good time to stop.

5.16.5 Brainstorming for a Virtual Team

Brainstorming does not have to be done in real time. For a virtual team, brainstorming can be accomplished by using online collaboration tools, but it is also a good option to brainstorm using bulletin boards and online postings. This is especially true when the team is spread over different time zones across the globe. There still must be a facilitator and the topic of the brainstorming must be clearly stated (this may be even more important in a virtual environment). A simple noninteractive brainstorming approach for a virtual team is

1. Send out the brainstorming statement and get agreement from the team.
2. Ask participants to post their initial ideas on a bulletin board. Set a deadline.
3. Close the bulletin board and allow for review time.
4. Open the board for add-on ideas.
5. Repeat steps 3 and 4 again (this can be done many times, but it is good to set a time limit for consideration).
6. Review for duplication and see if there is consensus to remove any ideas.
7. Categorize (possibly using a top-down affinity diagram) and document.

5.16.6 Summary

Advantages of brainstorming:

- It generates a large number of ideas in a short time.
- It encourages creativity.
- It allows for group synergy.

Disadvantages of brainstorming:

- It is possible to lose control to strong personalities (*group think*).
- It is difficult to get the team to stay inside of boundaries.
- It can generate conflict.

Best practices:

- Create an environment of acceptance.
- Everyone's input is valued.
- Group ideas into categories.

5.17 Tools for Virtual Environments

Each one of the requirements-gathering techniques just reviewed may need to be done in some form of virtual environment. That environment may range from something as simple as a phone interview with the customer in the next building to something as complex as a JAD session with participants from three continents through video-conferencing and involving translators. Perhaps the most important aspect of being successful in a virtual environment is to realize that there will be problems and there needs to be a well-thought-out plan to deal with them.

Some of the tools that may be used in the virtual environment are old existing tools, some are new and advanced. The drivers behind what to use are

- Cost. Videoconferencing and such tools can be very expensive, especially if the participants are spread across the globe. However, such tools do give the advantage of being able to see the other people in the session, to read body language, and to build a sense of team. It is a good idea to try to do the first meeting in person because that can make future videoconferencing sessions more productive.
- E-mail. Using e-mail, although common and accessible for most organizations, also can cause some difficulties. Response is slowed down. Often, there is a tendency for overcommunication, where the e-mail goes out to too many people. Limit the distribution lists and stay focused in the e-mails.
- File sharing, collaboration tools, and project rooms on the web. These areas are evolving fast, and there are a number of places that can be used to set up virtual groups where members of the group can share information, have discussions, post questions, etc. There are also tools such as Wiki and blogs, which are often free, easy to set up, and easy to use for sharing information and performing collaborative tasks. Be aware, though, of the level of technological maturity among the stakeholders. For some people, using blogs and spending time online is second nature, but for many it is not. Seriously consider if the organization is ready for these tools.

5.18 Requirements Prioritization Techniques

Once the requirements have been captured, there is often a need to prioritize them, either because the budget and timeline do not allow for all requirements to be incorporated or to help evaluate alternative solutions. To make these decisions, it is important to know which features the customer needs the most. Quite often, a product is planned to be released in multiple stages, and prioritization can help determine what features and functionality will be included in each of the releases.

Quite often a product is planned to be released in multiple stages, and prioritization can help determine what features and functionality will be included in each of the releases.

The easiest way to prioritize is to ask the customer what is most important. However, the standard answer is usually, “Everything is important!” So over the years, various techniques have evolved to get the customer to participate in the prioritization process and to make it as objective as possible. There are four prioritization techniques that will be described here:

- Dollar approach
- Forced-pair ranking
- Density dotting
- Analytical hierarchy process (AHP)

Each has advantages and disadvantages; the first three are relatively simple, the last more complex. It is recommended that you pick one approach and use that approach on all projects. Prioritization techniques can get very confusing, and they really work best after the customer has used them a few times.

5.18.1 Dollar Approach

This approach is easy in concept, but sometimes difficult to put into reality. It is based on putting a price tag on each of the key features of the product. Stakeholders then get an amount of money to spend and can each go on a shopping spree. They can distribute the money any way they see fit, but they cannot overspend. [Figure 5.9](#) shows how this may look in an environment with three stakeholders.

The total cost of the features shown is \$750. Each department is given a budget of \$450 and is told to spend that money any way they want. On the basis of their responses, feature 1 got three votes and features 3, 5, 6, and 7 each got two votes. The method can be tweaked for each situation, but the basic premise is that the customers vote with their wallets.

There are some key considerations that must be looked at with this technique:

Feature	Cost	Marketing	Operation	Service	Total
1. Cross store synchronize	\$100	X	X	X	3
2. FDA link	\$200		X		1
3. Manufacturing link	\$100	X	X		2
4. Patient analysis	\$50			X	1
5. Drug analysis	\$50		X	X	2
6. Wireless access	\$100	X		X	2
7. Web access	\$150	X		X	2
	\$750	\$450	\$450	\$450	

Figure 5.9 Dollar prioritization.

- Is there enough information to estimate the cost? This is typically done in the early stages of the project, and because there has been no real evaluation of solutions, any estimate will be crude at best.
- Consider using relative rather than real numbers. Use \$100, \$200, and \$500. The customer knows it is not the real number. The intention is to show that some of the features are much more complex than others.
- Not all stakeholders have the same priority. If one stakeholder has a higher priority than the others, then this technique does not work well.

5.18.2 Forced-Pair Ranking

Forced-pair ranking is probably the simplest and most common technique for prioritizing requirements. All requirements or features are compared against all other requirements in a pair, and the most important one is selected. It works very well in an environment in which there are many stakeholders with the same set of priorities. The first step in forced-pair ranking is to build the ranking matrix (which takes some training), as shown in [Figure 5.10](#).

With this structure in front of the group, each requirement is compared against each of the other requirements. The group is asked, “Which is more important: FDA interface or manufacturer interface?” In our sample team of five people, there were three votes for manufacturer and two for FDA. Note that everyone must vote. Place the number 3 (the votes for the manufacturer) below the diagonal line in the box that intersects manufacturer interface and FDA interface. Place the number 2 above the line, closer to FDA. Go through the same process for all the other requirements. At the end, the diagram should look like [Figure 5.11](#).

Next, add up the numbers for each requirement:

- FDA interface = $2 + 1 + 3 = 6$
- Manufacturer interface = $3 + 5 + 4 = 12$

1. FDA interface	1. FDA interface		
2. Mfg interface		2. Mfg interface	
3. Wireless			3. Wireless
4. Web access			4. Web access

Figure 5.10 Forced pair.

1. FDA interface	1. FDA interface			
2. Mfg interface	2	2. Mfg interface		
3. Wireless	1	5	3. Wireless	
4. Web access	3	4	4	4. Web access
	2	1	1	

Figure 5.11 Forced pair filled in.

- Wireless = $4 + 0 + 4 = 8$
- Web access = $2 + 1 + 1 = 4$

On the basis of this, we can see that manufacturer interface is the most important requirement, three times as important as web access. This technique does take some time to get used to and most people struggle a little with finding the right numbers to add, so make sure to practice this before doing it with the customer.

5.18.3 Density Dotting

Another approach to prioritization, sometimes used in agile development, is density dotting. It is less formal than other approaches and can be used for a number of scenarios. In a facilitated session, there may be ten features identified and it may be clear that there will not be time to discuss all the features. Have the participants use a marker and flip chart to put a dot next to the top three things that they wish to discuss and explore in more detail (or that they view as the top priorities). Density dotting will, often in less than 60 seconds, produce a visual representation of what the group wants to discuss, and by extension, where the most important features and requirements are that will be explored. The top three (or five, or whatever number there is time for) will then be the focus for a more detailed discussion.

Density dotting will, often in less than 60 seconds, produce a visual representation of what the group wants to discuss.

This type of approach removes a lot of the tension from the prioritization and decision process. It is simple, equitable, and fast. However, it does assume that everyone on the team has the same power when it comes to deciding what to focus on, which may not be true. For instance, if on the Prescription Interaction Project the team decided that the top three topics to discuss were (1) entering customer

information, (2) identifying previous prescriptions that may cause a bad interaction with the current one, and (3) overriding drug warnings, other topics, such as “checking against family members’ prescriptions,” could be overlooked because only the sponsor and the lawyer wanted to talk about them. So before doing density dotting (or any prioritization effort), make sure that there is buy-in from all stakeholders.

5.18.4 Analytical Hierarchy Process

The AHP is very similar to the forced-pair method, but it adds the question, “How much more important is requirement A versus B?” In the forced-pair ranking approach, the question simply asks which is most important. The AHP approach takes more time because the customer may have to do some analysis to know how much more important one feature is versus another. Part of this thought process should be the relative expense of the feature. Because this prioritization is typically done very early in the process, it is likely that the benefit as well as the cost estimate will be at a high level. That is OK. The customer does need to know if this is a \$1,000 requirement or a \$10,000 one.

5.18.5 Summary

Advantages of prioritization:

- It allows for implementing the most important areas first.
- It gives information for trade-off between requirements.
- It can be used to evaluate change requests.

Disadvantages of prioritization:

- It can be very cumbersome and still be fairly subjective.

Best practices:

- The customers must set the priorities.
- If using to evaluate different solutions, only look at requirements that will vary based on the solution. If a requirement is met by all possible solutions, then there is no need to prioritize it. However, if the prioritization is to determine what to build and what not to build, then all the requirements should be prioritized.

5.19 Prepare Requirements Package

Table 5.2 is an example that shows the main stakeholders for a BRD and what their main focus areas might be from the requirements package.

Table 5.2 Stakeholder Focus for the BRD

STAKEHOLDER	MAIN FOCUS AREAS
Customer	Regulatory requirements
	Business requirements
	Assumption
	Constraints
	Risks
User	User requirements
	System requirements
	Business rules
Developer	All
Executives	Goals and objectives
	Impacts
	Risk
	Assumptions
Internal organizations	Constraints
	Process changes
	Interface changes
External organizations	Regulatory requirements

- Customer: The primary concerns of the customer are the business requirements, regulatory requirements, assumptions, constraints, and risks. Customer sign-off on the BRD is not going to be based on the detailed user interfaces or the system response time. Those requirements are likely to be left for the actual users of the system.
- User: As long as the customer (the management) is happy with the high-level requirements, the users are unlikely to be concerned about them. Rather, they would be more interested in the usability, efficiency, learning curves, accuracy of data, and other items that are likely to have an impact on their day-to-day job activities. The users are often under time pressure, so if they are given too much information, they may not have time to review what will impact them the most.
- Development team: This is the only stakeholder to whom it is recommended that all of the information, the complete BRD, be presented and reviewed. Because this is the team that will take the requirements and create a solution, the development team must have a complete understanding of all categories of requirements, the as-is as well as the to-be business and system situations, along with information about user profiles, assumptions, constraints, and so on. If the system analyst does not have a clear understanding of both the big picture and the detailed requirements, chances are that the solution will miss the mark.

If the system analyst does not have a clear understanding of both the big picture and the detailed requirements, chances are that the solution will miss the mark.

- Executives: They are probably the ones hardest to communicate with, or rather to judge how much information to communicate to them. They must be made aware of goals and objectives, impacts to the business, risks to the organization, and any high-visibility assumptions and constraints. The communication to the executives should be concise and allow them to ask questions if they need more information. Also look at each item and see if it is really something that should be reviewed by the executives. Although undercommunication is potentially very harmful to the organization, overcommunication can also be a serious problem. Stay focused, make it short, and make it clear.
- Other internal organizations: Here, the communication effort should be about changes to business processes and workflow that may impact other areas. A special highlight should be on the gap analysis between what is done today and what will be done in the future.
- External organizations: There may not be a need to communicate with them at all, but if the system being worked on interfaces with vendors or external customers, there should be some communication, albeit high level, to make them aware of what is going on. They may also need to be aware of any regulatory requirements that might impact them.

When creating the packages that will be reviewed with the stakeholders, consider both their need for detail and their need for a big-picture understanding. Some of the most common tools to allow for this are

- Models—Some different modeling techniques are reviewed in [Chapter 8](#). Although models are often used to effectively elicit requirements, they can also be just as effective when used to communicate the BRD to a large audience. Some of these diagrams are very intuitive and help give the reviewer a quick, high-level view of what the system is all about. Models are also great in meetings where the discussion can stay focused by highlighting one area of the model at a time.
- Text—Can be documented in templates or in matrices. Text can often be overwhelming, especially on large projects, so it is important that it is well organized and grouped together based on a well-defined taxonomy (which is discussed in [Chapter 8](#)). To make it less overwhelming, try to present the text within the context of a model whenever possible.
- Prototyping—While technically a type of a model, the prototype will allow the simulation of predefined aspects of the system, and is often the most powerful way of reviewing requirements. The business analyst must make sure that the focus is on the requirements of the solution rather than the solution itself. The more highly functional the prototype is, the more difficult it will be for the user to stay focused at a high level.

The *BABOK* published by the IIBA also discusses the need to add material to the BRD review that is outside of the actual project documentation. Two especially

powerful ways to help ensure that the reviewers understand the material being reviewed are through the use of presentations and storyboards. Presentations can help explain the rationale and background of the project, as well as the requirements for the solution. A good presentation will highlight the most important information and make it easy to understand. For example, when reviewing the BRD with management, it can best be done by a presentation containing

- Business problem and vision
- Project boundaries
- Impact of solution
- High-level assumptions and constraints
- Business-level requirements
- Key quality-of-service requirements
- Any controversial requirements

By putting the right focuses on the review, what could have been many hours of nonproductive review can now be brief and to the point.

The second way to help with the reviews is to use storyboards (or user stories, which are what the IIBA calls them). Storyboards are discussed in earlier in this chapter as a way to gather requirements. Here, they are used as a way to explain to the stakeholders what the system will do. Storyboards can be used for reviews even if they were not used to elicit requirements. So instead of using a thick requirements document to explain all the requirements of the Prescription Interaction Project, a storyboard is used to show what happens when a customer walks into the pharmacy with a prescription for a drug that has a dangerous interaction with another drug the customer has bought in the past. Each step is shown on the *board* (which can be a slide in a PowerPoint presentation). At the draft level, it may look something like the storyboard in [Figure 5.12](#).

The review becomes an intuitive way to discuss a series of steps:

1. The customer presents a prescription.
2. The pharmacist enters the information.
3. The system searches for potential drug interactions.
4. The system searches the customer record for interacting drugs.
5. The system prompts the pharmacist to ask if the customer is currently taking any other of the interacting drugs.
6. If the customer presents a drug interaction override approval, the pharmacist verifies with the doctor.
7. The pharmacist accepts or rejects the prescription.

The business analyst must plan for sufficient time during the analysis phase to create these presentations and storyboards. Although it may be difficult to know the extent to which these will be needed, by evaluating the complexity of the project and

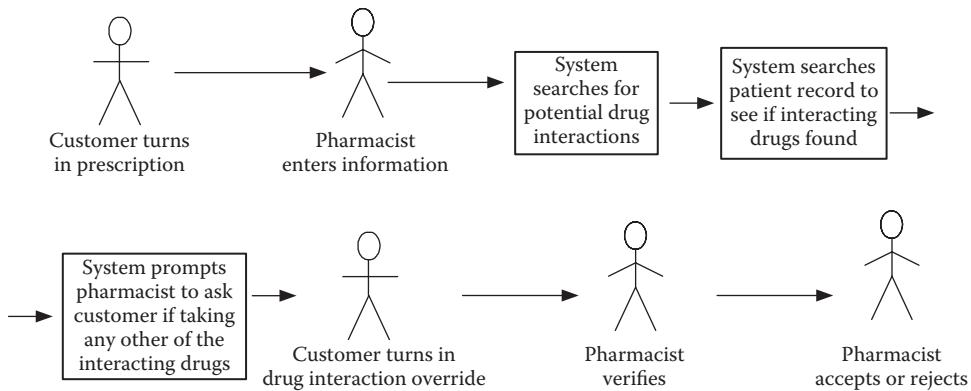


Figure 5.12 Storyboard for communication.

the diversity of the stakeholder base, the analyst can make an adequate guess about how much effort to plan for.

Once the determination has been made about who needs to review the package, as well as what information each stakeholder should review, verify this with the stakeholders and get an agreement on the package. Depending on the organization, the sponsor will sometimes be the person to determine what a certain stakeholder should see; however, whenever possible, it is best to review this with the actual stakeholder. This may already have been decided and may be a part of the communication plan for the project or for the analysis phase. If so, just grab the information from there.

The formality of the package presented will depend on whether it is reviewed by an internal or external reviewer. The stakeholder's location can drive how formal the package needs to be. If the reviewer is a development group located somewhere in Asia, the package will need to be much more complete and able to stand on its own than if the developers are sitting right next to the business analyst. Different types of projects will also require different degrees of formality in the review package. The IIBA has identified four types of projects that will require different types of review packages:

1. Custom-developed software: The review will need to be very comprehensive and detailed. For the most part, the review will be based on the models and text in the BRD.
2. Upgrading to new technology: If the business does not change, then this may be primarily a technical review with limited customer involvement. The customer would primarily need to know what will work differently.
3. Change in a business process: This will need a review of the enterprise impact and business rules, as well as reviews of process and data models.
4. Purchasing commercial off-the-shelf software: The review may be focused on the business and user requirements as well as constraints. The detailed functional requirements will often be built into the system, and it will be more a matter of explaining to the user how the package works. Part of the review may be to bring in the package, allowing the reviewers to see what they will

get. Be aware, though, that the requirements review session should never be the first time that a stakeholder sees the package; the stakeholder would be likely to start treating that as a requirements-gathering session rather than a review session.

Expanding on the last detail, I can relate a real-life experience about the dangers of assumptions when reviewing requirements. When developing the Saturn dealer system, there had been a number of requirements sessions, including prototypes and text documents. After the main part of the development was done, the dealers were brought back one more time to see what they had created and give it their blessing. Unfortunately, now they had a different opinion. It quickly turned into another requirements session resulting in massive changes to the project. So, make sure that the right level of expectation is set before starting the review.

Most of the review package will be excerpts from the BRD, but there needs to be an introduction section added for each package. This introduction section should include

- Overview of package
- Timing and location of review meetings
- Expectations of what the stakeholder should do with the package (glance at, analyze, ask questions about, approve)
- List of other reviewers
- Steps and deliverables in the review process

Having this package before a review meeting will increase the chance of having a prepared stakeholder going into the session. Although there is no guarantee that the stakeholders will read the package, at least they have been given the opportunity to do so.

5.20 Communicate Requirements

Whether this is done as a self-guided review by the stakeholders or in a presentation-type format, it is important to guide the stakeholders through the information. If a model is reviewed, do it step by step and focus on one piece of the model at a time. Verify that there is agreement with each focus area as the review is progressing. It is very difficult for the stakeholders to get comfortable with approving a large review package unless they can see the pieces one at a time and understand them. If the stakeholders get overwhelmed in the review, they are likely to push back and try to delay approval.

[Figure 5.13](#) shows an example of a data model being reviewed with the customer and demonstrates the way the review is focusing on one area of the model by shading the other areas. Once this area has been approved, change the focus to a different part of the model and review that. This also allows the reviewers to pay extra close attention to the areas that are most important to them. As an example, if this is a review

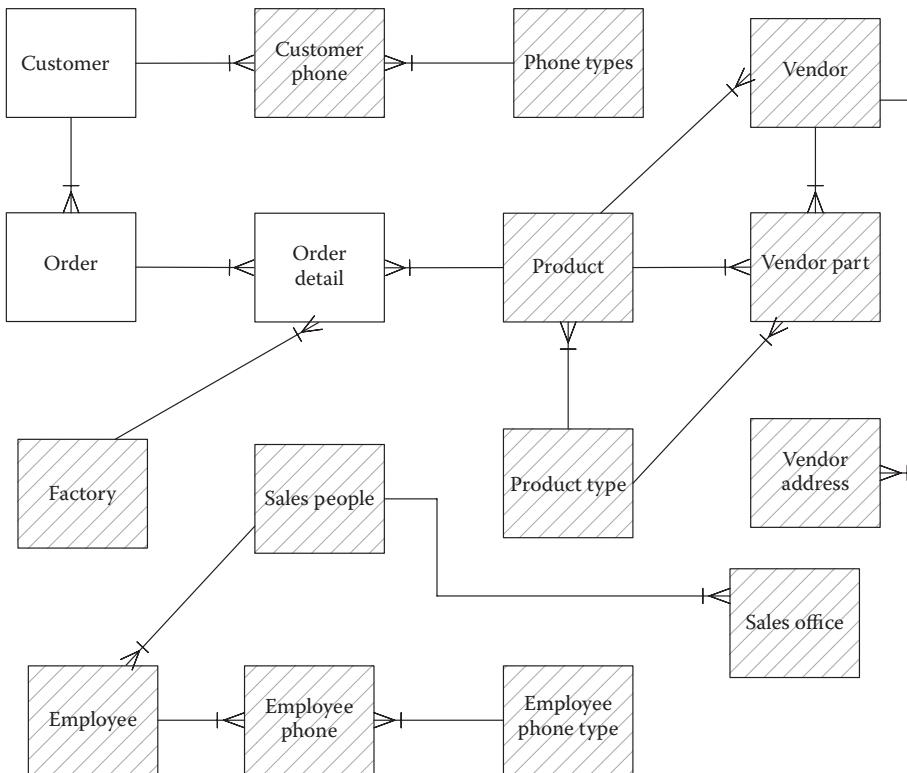


Figure 5.13 Data model for review.

for the prescription interaction system and the pharmacists are the stakeholders, they are likely to focus on what information is needed for the actual prescription. When reviewing doctor information or the patient information, the pharmacists may be less interested because it may not be a critical part of their job function.

A similar approach can be used when reviewing use cases. Clearly highlight which use case is being reviewed and do a complete review of that use case before moving on to the next. It is common for the stakeholder to start talking about areas of the system other than the part being reviewed. Avoid the temptation to jump over to that other area unless it is clearly essential for the current discussion. A good review takes good facilitation and part of that is to keep the review process focused.

5.21 Summary

The selection of the best requirements-elicitation techniques is difficult but critical to the success of the analysis phase. Do not consider only the skills and personalities of the users, managers, and other key stakeholders, but also review the skills of the business analysts. It may be that a facilitated session would be the best way to capture requirements from a customer, but if there is no experience or skill, it may be better to select a different review process. The approach that is selected must be validated with

the sponsor and other key stakeholders. Their time will be impacted, so their commitment is needed. Also evaluate stakeholder priorities, especially as requirements prioritization is taking place.

Activity

Take another look at the case study in [Chapter 10](#). Identify all the stakeholders from whom requirements will be elicited. Some of these stakeholders may have been identified in the activity for [Chapter 4](#). Include those but also add any others that come to mind.

1. Identify the best requirements-elicitation technique for each stakeholder on the list and document what actions the analyst should take to make the effort more successful with each stakeholder. This includes actions to take before requirements are gathered, during the requirements-gathering stage, and as follow-up actions.
2. Select one area where a discovery session would work well. Identify which stakeholders would be invited to the session and create an agenda for what to cover in that discovery session.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

6

REQUIREMENTS LIFE CYCLE MANAGEMENT

England and America are two countries divided by a common language.

George Bernard Shaw

This chapter concentrates on the management of the requirements-gathering effort.

From a management view, this knowledge area deals with understanding changes and how they impact the requirements. It also looks at how requirements can be reused by the organization in future endeavors.

6.1 Objectives

- Create traceability from project scope back to business need and forward to the solution.
- Identify stakeholders who need to be aware of the project direction.
- Work with people and organizations that have conflicting views on the direction of the project.
- Get formal sign-off for the requirements.
- Deal with changes to the requirements

6.2 Overview

Although the business requirements document (BRD) should be a complete and comprehensive document used for communication with stakeholders, this does not mean that every stakeholder should be expected to read every part of the BRD. The project manager must understand anything that impacts overall scope, timeline, and budget, but does not need to see detailed requirements. The executives and the sponsor must understand all of the business requirements and the business impact of the product, but they may not need to see the detailed user interface requirements. Too often, the stakeholder is handed a document hundreds of pages thick, and asked to review and approve the package. That is an unrealistic expectation.

The BRD review should be customized to fit each stakeholder. It is a good guideline for the business analysts to ask themselves with regard to each piece of documentation presented to the stakeholder, “Why is this stakeholder reviewing this document?” If the answer is something like “For information” or “To keep the stakeholder in

the loop,” then consider excluding it from the review package for that stakeholder. However, let the stakeholder know that the information is available; the stakeholder will choose to review it if interested. It is always best to let the stakeholders know when there is information excluded from their package.

6.3 IIBA Tasks

Requirements life cycle management is made up of five tasks:

1. Trace requirements.
2. Maintain requirements.
3. Prioritize requirements.
4. Assess requirements changes.
5. Approve requirements.

6.3.1 *Trace Requirements*

Requirements traceability is a two-way process. Requirements must be traced back to business objectives and forward to a product. [Figure 6.1](#) shows an example of how traceability can be shown.

It is important to note on this figure that traceability does not only mean that the requirements can be traced back to an objective but also that the objective can be traced forward so that it can be verified that the solution is enough to meet the objectives. In other words, we must show coverage of the higher levels in the figures as the lower levels are documented.

Planning for requirements management is discussed in [Chapter 4](#). The task of managing solution scope and requirements is related but focuses on how changes are being communicated, and how issues and conflicts that arise regarding different views on the solution scope are handled.

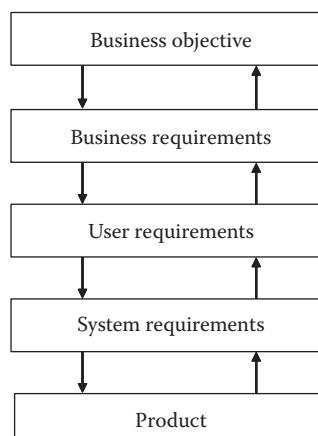


Figure 6.1 Traceability.

It is common in a multi-stakeholder environment for there to be disagreement between stakeholders regarding requirements, especially in the areas of project scope and functionality. What should be in and out of the scope of the product? One stakeholder may prefer more manual processing, while another prefers automation. One stakeholder may prefer outsourcing of a business function; another may want to handle it within the current project. It is not the business analyst's job (nor the project manager's) to determine what is in scope and out of scope. As discussed in [Chapter 4](#), a stakeholder analysis should be conducted and each stakeholder prioritized. If there is a difference in priority between the stakeholders, then this is easy: let the stakeholder whose concern has the higher priority decide. However, often there is no difference in priority. When this type of disagreement occurs, work with the different parties to try to reach a decision. If that does not work, then look at escalating it to the sponsor.

It is not the business analyst's job (nor the project manager's) to determine what is in scope and out of scope.

There is rarely just one way of performing a job function. Two different users may perform the same job two different ways in today's environment, and they may both feel that their way is the best one. This kind of disagreement can be handled through a process like the one just described; however, another possible solution might be to build enough flexibility into the product to allow both parties to keep operating their way. This is likely to increase the overall cost of the project (and potentially of the product), so try to minimize the use of this option as a solution.

6.3.2 Maintain Requirements

In an organization, new initiatives are often started by developing information that has already been created for earlier initiatives. In other words, we are recreating documents. To save time and ensure better consistency, it is a best practice for an organization to identify those requirements that should be used for future initiatives and to have a repository where they are stored. When doing this, care must be taken to maintain the integrity of those requirements and to update them as the business changes.

6.3.3 Prioritize Requirements

When developing a list of requirements, it is often necessary to determine the value and importance of those requirements. This will help determine which requirements should be implemented first, as well as give us an idea where to focus our efforts. There are many factors that drive priorities, among them:

- Business value
- Risk

- Difficulty
- Regulatory compliance

In [Chapter 5](#), different techniques for prioritization of requirements are covered. Specifically mentioned in the *BABOK®* Guide is MoSCoW analysis, which sorts requirements into

- Must
- Should
- Could
- Won't

6.3.4 Assess Requirements Changes

Change control was discussed in [Chapter 4](#). There is typically a different change control board for product scope changes compared with project scope changes. The business analyst tends to be the owner of preparing and evaluating the impact analysis of product changes.

6.3.5 Approve Requirements

Planning for requirements management was also discussed in [Chapter 4](#). The task of managing solution scope and requirements is related but focuses on how changes are being communicated and how issues and conflicts that arise regarding different views on the solution scope are handled.

It is common in a multi-stakeholder environment for there to be disagreement between stakeholders regarding requirements, especially in the areas of project scope and functionality. What should be in and out of the scope of the product? One stakeholder may prefer more manual processing, while another prefers automation. One stakeholder may prefer outsourcing of a business function; another may want to handle it within the current project. It is not the business analyst's job (nor the project manager's) to determine what is in scope and out of scope. As discussed in [Chapter 4](#), a stakeholder analysis should be conducted and each stakeholder prioritized. If there is a difference in priority between the stakeholders, then this is easy: let the stakeholder whose concern has the higher priority decide. However, often, there is no difference in priority. When this type of disagreement occurs, work with the different parties to try to reach a decision. If that does not work, then look at escalating it to the sponsor.

It is not the business analyst's job (nor the project manager's) to determine what is in scope and out of scope.

There is rarely just one way of performing a job function. Two different users may perform the same job two different ways in today's environment, and they may both

feel that their way is the best one. This kind of disagreement can be handled through a process like the one just described, but another possible solution might be to build enough flexibility into the product to allow both parties to keep operating their way. This is likely to increase the overall cost of the project (and potentially of the product), so try to minimize the use of this option as a solution.

6.4 Summary

Identify who should review the requirements and the best way to present the information to them. Doing this greatly increases the chance of achieving user acceptance of the product down the road and minimizes the risk of the users saying, “I never knew that.”

Activity

1. Document what components you would include for a BRD review with the following case study stakeholders:
 - Sponsor
 - Distribution center manager
 - Vice president of operations
 - Buyer
 - Developer
2. Document what each of the stakeholder’s main areas of focus should be for the requirements package being developed.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

STRATEGY ANALYSIS

Everybody gets so much information all day long that they lose their common sense.

Gertrude Stein

Strategy analysis is perhaps the fastest-growing area of business analysis. It provides the context and the link between projects and the business. Although most of the business analyst's work tends to reside within the project boundaries, there are parts of the job that take place before and after the project. This section deals with most of the pre-project activities, such as feasibility studies and industry benchmarking. It is often the most difficult part of the business analyst's job because it can be highly political, complex, and unchartered. The pre-project activities tend to give the analyst the most organizational visibility through dealings with the upper levels of the organization. If the business analyst is successful in this area, it can be a great career enhancer; if unsuccessful, it will be noticed.

7.1 Objectives

- Understand the importance of the business analyst having a strong understanding of the enterprise.
- Review ways of documenting and understanding the business.
- Identify business vision, goals, and objectives.
- Identify prerequisites to requirements gathering.
- Identify how a project maps to the enterprise.
- Document a strategy analysis for the case study.

7.2 Overview

The International Institute of Business Analysis™ (IIBA™) defines strategy analysis, in the *Guide to the Business Analysis Body of Knowledge® (BABOK®)* version 3, as the knowledge area that describes the business analysis activities needed to

- Identify a business need
- Assist the enterprise with addressing the need
- Align the initiative with other initiatives in the enterprise

This chapter reviews this knowledge area and also discusses the skills the business analyst must have to accomplish this work. It is important to note that this knowledge area may be the one to change the most over the next few years. Strategy analysis is not the traditional role of the business analyst, and the type of work done in this area varies greatly from organization to organization. Right now, the domain of the business analyst overlaps in areas with project management, program management, and business architects. As the domain of business analysis becomes more refined over the next few years, some of the overlapping areas will become clarified and some will not, requiring each organization to customize its definition of the business analyst. This tailoring process is still going on within most project management organizations more than 30 years after the Project Management Institute™ (PMI™) was formed, so it will take some patience for the IIBA as well.

Strategy analysis is not the traditional role of the business analyst, and the type of work done in this area varies greatly from organization to organization.

Although the primary role of the business analyst is to function as a communication link between the customer and the developer in relation to defining the requirements of a product or service, the role is expanding in many organizations. Clearly, the business analyst must have a solid understanding of the overall business environment in which the customer operates. The business analyst is often assigned to do competitive benchmarking against other companies, evaluate best practices, and in general keep up on trends in the industry. This naturally makes the necessity for a strong business background an increasing demand on future business analysts.

Some areas of involvement for the business analyst, in addition to the areas outlined by the IIBA, may be

- Benchmarking: This can be part of a feasibility study, but can also be a stand-alone activity. The purpose of benchmarking is to evaluate where the customer's competitors are. Who is considered to be *best in class*, who is improving rapidly, and what new trends are emerging in the customer's area of business. For the C.V. Green Prescription Interaction Project described in [Chapter 1](#), this involves researching what the competition is doing, and one may need to be creative to do this. For example, it may be obvious to check out the pharmacies that are in the same business, but you should also look beyond the obvious and see if there are other industries in which similar work is done. There could be lessons to be learned from order entry projects or from medical billing.
- Identifying and analyzing new business opportunities: By attending conferences as well as being aware of and up to date on new technology trends, the business analyst can actually be the instigator of new opportunities. A business must continuously evolve and change to stay competitive, and the business analyst can be a catalyst in that process.

7.3 IIBA Tasks

The key tasks outlined by the IIBA for strategy analysis are

- Analyze current state.
- Define future state.
- Assess risk.
- Define change strategy.

These are the key tasks introduced by the IIBA for the knowledge area of strategy analysis. Each organization must determine which ones are applicable and to what level of detail each will be explored. Clearly, some organizations already have existing functions and professionals that have done this work for a long time and with great results. They may be the owners, the strategists, or the executives of the organization. That's great! All the business analyst needs to do is to trace the project objectives back to the deliverables, which may be the strategic plans, the tactical initiatives, or the company vision. However, for organizations that currently do not have a good handle on these activities, the business analyst may be the natural person to start taking on these responsibilities.

For organizations that currently do not have a good handle on these activities, the business analyst may be the natural person to start taking on these responsibilities.

Strategy analysis should not be viewed as a *one-size-fits-all* approach. There should be established ground rules regarding the level of effort needed for different types of projects and organizations. This customization may be driven by project size, risk, complexity, visibility, or any other parameter that makes sense to the business. As an example, if the customer submits a request to change a report or to implement a minor change in an existing system, there is probably no need to do an extensive strategy analysis. On the other hand, if the organization is switching to a new enterprise resource planning system, then the strategy analysis may be more extensive than the actual projects.

For many of the IIBA key processes, the inputs come from customers, government, or executives during one-on-one interviews, as well as from documents such as the strategic plans and goals of the business, the problems and opportunities that the business is facing, and regulations the business must follow. Although sometimes these documents are identified by executives during off-site planning sessions and at a level of the organization to which the business analyst does not normally have access, it is a good idea to have the business analyst as informed and involved as possible in these events. A strategic plan will lead to tactical initiatives within the organization. These initiatives potentially will lead to many projects, which will need to be prioritized on the basis of their ability to meet the strategic plan.

So what is the business analyst's role in creating the strategic plan and documenting the strategic goals? Most organizations will primarily use the management team

and often some consultants to facilitate and drive these efforts. However, this is an area where the IIBA appears to be visualizing a bigger involvement in the future for the business analyst. Once the organization develops business analysts with a strong understanding of the business along with good communication and facilitation skills, those analysts can become very effective in leading the sessions where the strategic goals are set. Because the analysts will become heavily involved with the implementation of the strategic plans, at least at a project level, the skills and knowledge obtained in this early effort will not be lost. However, sometimes strategic sessions will not be taken seriously if they are facilitated by an internal analyst. There is an old saying that no one is a prophet at home. If there are any concerns about this, then the organization would be better off bringing in an independent consultant to lead the activity. Even in this case, it is desirable to have the business analyst involved, either as a scribe, observer, or subject-matter expert.

An especially value-added role for business analysts in the creation of the strategic plan, regardless if they are actually in the sessions, is the creation of score cards and metrics. Any good strategic plan or set of strategic goals must be measurable to be effective. So what are the metrics needed to measure a strategic plan? They could be cost or profit related. Most strategic goals have at least some relevance to the finances of the business, such as market share or market position. C.V. Green wants to be the *premier pharmacy*. What does that mean? Highest market share? Most revenue? Most customers? Most satisfied customers? The business analysts are not the ones to answer these questions; rather, their job is to find the person who can answer them and then document the answers.

Without a clear understanding of this linkage between, and importance of, the different components of the strategic plan, it would be difficult to recommend potential improvements to the business. Returning to the Prescription Interaction Project, the strategic goal for C.V. Green is to become the premier pharmacy chain in the world through superior service and competitive cost. One tactical initiative is to minimize the harmful effects of drugs. This would provide benefits in terms of both competitive cost and the objective of superior service. Checking for prescription drug interaction (the project used as an example in this book) fits nicely here. By doing this trace and understanding how the project fits within the bigger business picture, the analyst can identify benefits, ask the right questions, and assist in getting stakeholder buy-in for the effort.

By learning about the global aspect of the goal of C.V. Green and the desire for superior service, it is likely the analyst will pay more attention to how to identify drug interactions as well as how to maximize customer safety versus just staying legally compliant. For instance, if the project had a purely legal driver, the focus would be narrower and only consider how the pharmacy could limit drug interaction based on best-available information. If the goal is primarily customer satisfaction and service, the business analyst may go out and look for what information could be obtained by working with other pharmacies, or by looking for drug interactions from drugs that the customer's family members have bought.

This ties in well with the topic of validation, which is discussed in more detail in [Chapter 8](#). The purpose of validation is to make sure that the product or service that is eventually developed will solve, at least partially, the business problem or opportunity that caused it to be started. Naturally, this is difficult to accomplish unless the organization as well as the business analyst clearly understand that problem or opportunity.

7.3.1 Define the Business Need

All projects should be based on a business need. Naturally, that need may exist at different levels. Some projects are strategic, some are tactical, and yet others are operational in nature. Defining that need is critical to the buy-in and success of the project.

There is an increasing need to put more emphasis on and resources into making sure that the business environment, vision, goals, and objectives are known and understood before a significant project is undertaken. Clearly, for many business analysts, this is beyond the scope of the work they do; however, for those who get assigned to large, enterprise-wide, high-visibility-type projects, knowledge of this area is crucial. In many organizations, this activity may primarily involve digging for information. The strategic goals and tactical initiatives are often already documented at a corporate level. The link between these and the projects done at the operational level is part of the traceability needed to justify the existence of a project. [Figure 7.1](#) is an example of a model of those relationships.

The executives may have an annual strategic session or there may have been consulting studies done on where the company is heading and its vision and mission. strengths–weaknesses–opportunities–threats (SWOT) analysis is a tool often used to show the status of a business. [Figure 7.2](#) shows an example of a SWOT analysis.

The SWOT analysis will often be a foundation for business strategic goals and plans. An aggressive business will focus on taking advantage of its strengths and exploring its opportunities. On the other hand, a risk-averse business may be more focused on

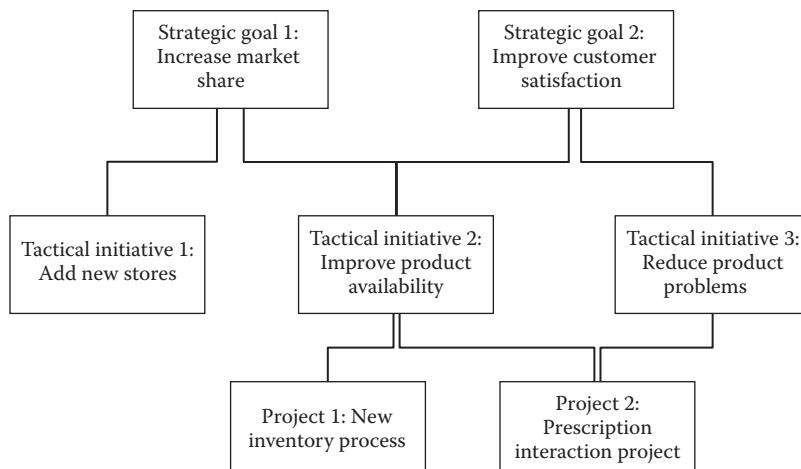


Figure 7.1 Strategic goals and tactical initiatives.

Strengths What are your competitive advantages? What are your strongest skills? What do you do well?	Opportunities How can you leverage your competitive advantage? How can you utilize your skills? How can you take advantage of what you do well?
<ul style="list-style-type: none"> • Superior technology integration • Strong product line • Good industry reputation locally • Strong sales force 	<ul style="list-style-type: none"> • Streamline and integrate business processes • Create local partnerships with global partners • Leverage current sales force globally
<ul style="list-style-type: none"> • Limited global presence • Lack of high tech networking • High cost of inventory 	<ul style="list-style-type: none"> • Competitor may have better global contacts • Communication between regions may be unreliable • Inventory cost may increase further with global expansion
Weaknesses Where are you lagging the competition? What skills do you lack? Where are your problem areas?	Threats How can the competition hurt you? Where can the competition surprise you? Where is the competition stronger than you?

Figure 7.2 SWOT analysis.

protecting against its weaknesses and defending against threats. Figure 7.2 shows that a key strength for C.V. Green is a superior technology integration process. However, a weakness is a limited global presence. A business would need to decide whether to focus on capitalizing on strengths or defending against weaknesses. Although it is quite possible to do both, in many cases this will lead to a dilution of effort, with the possible result that neither action is successful. What is the right approach? That is one of the many *it depends* questions asked in business analysis. The main driver is often the maturity of the organization and the personality of the decision makers. Is their attitude to utilize strengths and conquer? Or is it more of defense and survival? Those are the types of questions that may be answered during the review of a SWOT analysis.

By having this type of information available at the start of the requirements-gathering effort for a project, the business analyst can ask the right questions and be able to trace the decisions made for the project back to actual business needs. In addition to SWOT analysis, other aspects of business architecture are

- Policies and business rules—explain what standards the business has for certain actions. Policies and business rules tend to be overlapping. They both deal with actions that must be taken or rules that must be complied with by the

business in response to certain conditions. Examples include “The company will not accept a prescription unless payment is made or a preapproved drug plan is given,” or “the pharmacist must verify the doctor’s signature before issuing a prescription.”

- Procedures—show how certain key activities are carried out within the business. Examples of procedures can be *prescription refill procedure* or *drug interaction verification*.
- Competencies—show what the business should be capable of doing. They can include competitive advantages or unique skills such as “Ability to refill prescription from any store in the network,” or “Offer generic equivalents for 75 percent of all prescriptions.”

Although attention to these is necessary for the business analyst to be successful, it is important not to get them confused with actual requirements. A business must be able to follow its policies, procedures, and business rules and utilize its competencies within the developed system; however, the system itself does not explicitly have to enforce them. As an example, there may be a business rule that “a pharmacist must check for drug interaction before filling a prescription.” When the analyst is gathering requirements, this business rule should trigger some questions:

- Should this check be automated within the system?
- What information does the pharmacist need to be able to decide if there is harmful drug interaction?
- Does the pharmacist just need to check for it? If there is potentially harmful interaction with other drugs, what action should the pharmacist take?
- Are there any reporting or data requirements for this business rule?

A business must be able to follow its policies, procedures, and business rules and utilize its competencies within the developed system, but the system itself does not explicitly have to enforce them.

It is easy to understand the value of knowing the business rules, policies, procedures, and competencies when going into the analysis phase of the project. If they are not readily available, the business analyst may need to be a part of documenting them. This is part of the overall *as-is* documentation. In the business requirements document (BRD) templates shown in [Appendix B](#), the business rules and other similar artifacts are documented in the *as-is* section of the BRD. This section may not always be present in a BRD template because the *as-is* documentation often exists within other business documents or existing system documentation. In those cases, it is sufficient just to reference the source documents. Do not create duplicate information by copying information into the BRD.

There are existing modeling techniques available to describe a business. One that is mentioned by the IIBA is the Zachman framework for enterprise architecture, as shown in [Figure 7.3](#). This framework looks at different views of the business; for example,

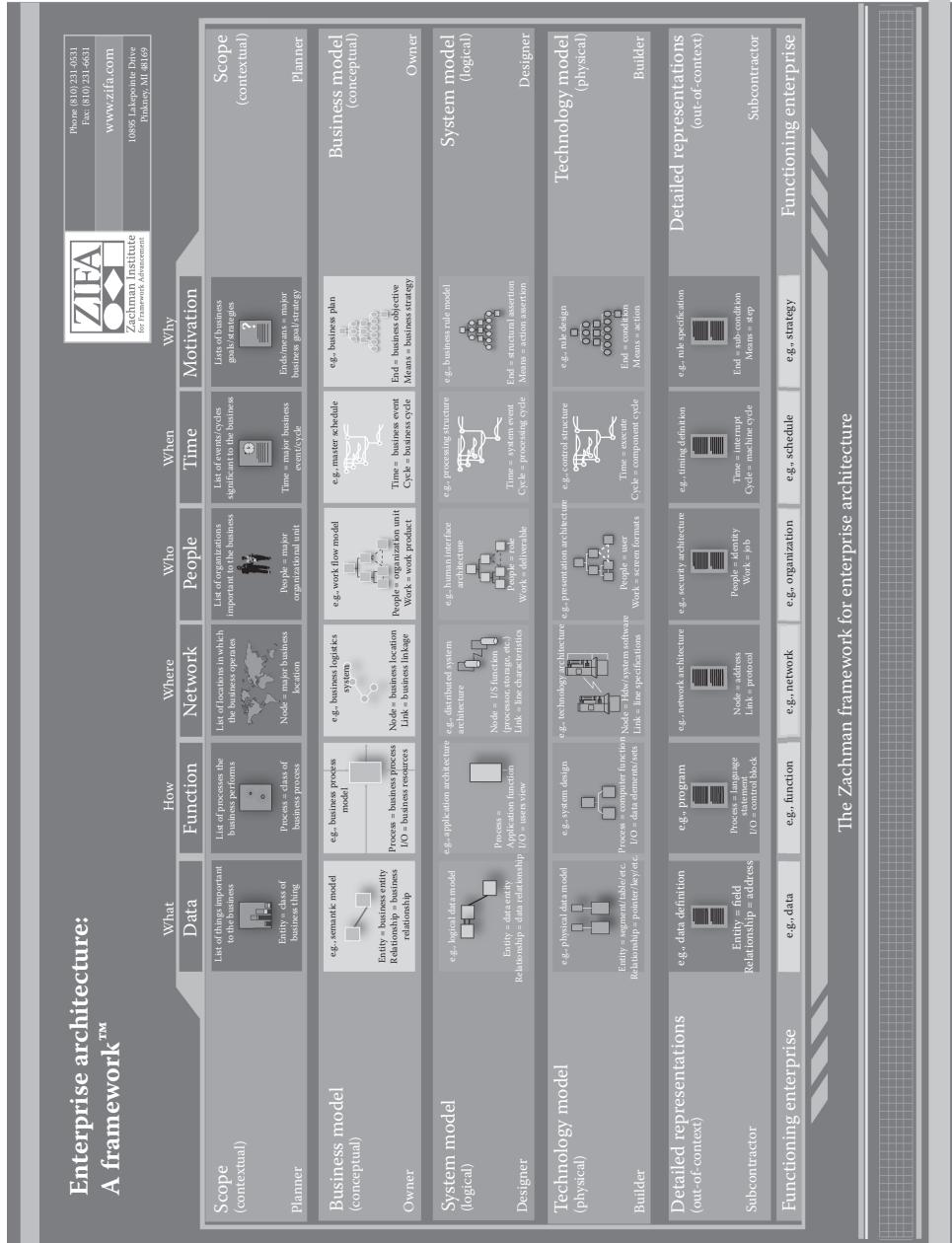


Figure 7.3 Zachman framework for enterprise architecture. (© John A. Zachman)

the business model, technology model, and system model. It then describes what, how, where, who, when, and why for each of those models, and identifies a specific deliverable for each intersection. As an example, the *why* for the business model is documented in the business plan, and the *what* for the system model is contained in the logical data model. It provides a very structured view of the business. There are, of course, other techniques besides Zachman, taking different approaches. The more a technique is used within an organization, the more comfortable everyone will be in viewing and analyzing the information. Rather than trying to find the perfect model, pick one that is usable and train the organization on how to use it. [Figure 7.4](#) shows a business model for the C.V. Green Pharmacy, tying back into the Zachman framework from above.

The *why* for C.V. Green is *premier pharmacy* and *cost-effective*. This can be looked at as the vision of the organization. The *what* is then at the mission level, including areas such as *efficient purchasing*, *global presence*, and *customer safety focus*. *Where* just shows the business's geographical locations, and *who* shows the organizational structure. *When* deals with business cycles and key business events, and *how* is what will link to the projects and initiatives going on in the organization. While this is obviously a simplification, it shows the basic thought process needed for a strategy analysis and understanding of the business model.

The activities of strategy analysis will necessitate the business analyst interfacing with executives and other stakeholders who are not necessarily impacted by any project being contemplated. If tasked with documenting the as-is situation for the business, the business analyst should start by doing a stakeholder analysis. It does not matter if a stakeholder analysis has already been done for the project. Describing the as-is situation of the business may need to go beyond the stakeholders already identified by the project manager. The question here is not just who will be impacted by the project, but rather who has information that may help the analyst understand the business. The marketing department at C.V. Green may not be involved directly with the Prescription Interaction Project; however, it does play a role in identifying the enterprise architecture within which the project exists.

An important skill set for doing this work is the business analyst's communication skills and ability to have clear, concise conversations with upper management.

Zachman framework	What	How	Where	Who	When	Why
Business model	Global presence Customer safety Efficient purchasing	Prescription interaction project Inventory management initiative	Europe Asia North America	Headquartered in the United States Regional offices in London and Tokyo	Largest pharmacy in the world by 2010	Become premier pharmacy Stay cost-effective

Figure 7.4 C.V. Green business model.

Although some people do this naturally, for most analysts this will take both education and practice. Training in negotiation, communication, and facilitation will be helpful. There is a need to be well prepared, have clear questions, and a clear goal in mind of what the interview with the executives should produce. Although there is no need to be intimidated, there is a need to be focused and respectful of the time commitment given. In large organizations with many business analysts, there is probably someone in the group who is more capable than the rest at interfacing with the executives. Select these analysts and prepare them well. Perform some practice sessions before going in front of the executives. Because it often makes people nervous to speak in front of upper management, it is important to come prepared with a lot of questions and have the pertinent facts at hand. When presenting a business case, an impact analysis, or a feasibility study, make sure that there is information to back up all numbers that are shown. Even if it is never used, it will make the analyst more comfortable, knowing that the information is there if needed. It will also increase the likelihood that the executives will buy into the presentation. It only takes one missed fact or one “I don’t know” to leave a trace of uncertainty that may take a long time and a lot of effort to erase.

7.3.2 Assess Capability Gaps

There are two main areas to this task: understanding the current capabilities and assessing new capability requirements. Most businesses have a constant need to change and improve to stay viable. Even if the business does not want to change, the world around it, the competition, and the marketplace are changing, driving a need to change and improve the business. That means a continuous update of vision, mission, goals, and objectives. To implement these changing directions, there is a constant need to assess what the organization is capable of doing, and a feasibility study is a way of helping with that. A feasibility study looks at feasibility from an organizational point of view. By evaluating different potential solutions to a business issue, a determination can be made as to what the impact is on the organization and also on the outside world. Feasibility does not necessarily mean “Is it doable?” It may just as often mean “Based on our current situation, is this option reasonable?” Many feasibility studies are initiated based on the SWOT analysis reviewed earlier in this chapter. The SWOT analysis lets the business know its strengths and weaknesses, opportunities, and threats. The feasibility study uses these areas as inputs into an evaluation of what the business can do. Does the organization have the capability to overcome a weakness? How realistic is it to capitalize on the organization’s strength?

Treat the feasibility study like a project. As with most projects, there are dependencies on other projects and the study may or may not lead to other efforts in the future. But for now, the feasibility study is a project in itself and, as such, should have requirements, deliverables, and a project plan. There is often a narrower question to answer in a feasibility study than the generic “Is this feasible?” It could be a technical

view, it could be a globalization view, or it could be an evaluation of the skills of the organization. The more precise the initial question asked, the better defined and, by extension, the better executed the feasibility study will be.

The feasibility study is a project in itself and, as such, should have requirements, deliverables, and a project plan.

The key steps when performing a feasibility study are

- Identify the problem or opportunity. Always start by understanding why this effort is being made. Whether traceable back to a SWOT analysis or initiated by some legislative change, understanding the drivers behind the request will help in understanding what areas to concentrate on. For example, if the feasibility study involves evaluating the possibility of sharing prescription information with other pharmacies for the purpose of evaluating drug interactions, it would help to know if this is undertaken to enhance the competitiveness of the organization or to minimize lawsuits.
- Understand the current situation. This can tie into understanding the previously discussed business architecture. Are there any partial solutions in place with this organization or anywhere else in the market? What infrastructure is already in place? What is the competitive situation and how will that change if we do, or do not do, the effort? A feasibility study is largely an analysis of “Where are we?” and “Where do we want to be?,” and determining what the gap is between those two and how the gap can be filled. Understanding the current situation gives the answer to “Where are we?”
- Define a vision of the solution. Following the thread from the previous bullet, this answers “Where do we want to be?” At this point, do not get bogged down with reality or constraints or requirements. Here, the main objective is to define what the world will look like when the opportunity has been realized or the problem solved.
- Determine alternative solutions. Be creative! If the first idea that comes up gets selected, there is a chance that eventually someone will think of a better way, which can cause major rework to the project. Alternative solutions not only help in coming up with a better solution, but they also help in reviewing and rejecting undesirable solutions early on in the process, which will save time and effort revisiting these later.
- Recommend a solution. Evaluate the alternatives and pick the one (or ones) that best addresses the original business problem. This can be done objectively, using financial data when available, or it can be done subjectively, by customer voting or the sponsor making a decision.

A big portion of feasibility is evaluating the risks associated with each of the alternatives. It is often necessary to take high risks to be successful and it is true that a

high risk may equal high payout; however, for the decision makers to make a well-informed choice, they must be aware of what the risks are. Conceptually, many of the techniques used for risk assessment are similar to the ones discussed in [Chapter 4](#), as well as later in this chapter in [Section 7.3.5](#), but at this level the focus is more on the organizational risk rather than the project risk. Capture both negative risks (threats) and positive risks (opportunities). In the end, the feasibility decision is largely based on comparing the value of the opportunities and the value of the threats, and if the opportunities are worth more, then the project will get a nod of approval.

Business analysts involved with doing feasibility studies must have broad business and information technology (IT) knowledge. They must also have a good understanding of financial analysis. For this type of effort, the analyst must be a person who can hone in on the important areas that will make a difference from the capabilities and financial standpoints. He or she must be able to understand the size of the investment needed and when it is needed, and to compare that to the benefits that will be achieved and when those benefits will be realized. This will include comparing the cost of developing a product in-house versus purchasing a ready-made product. This type of decision is often based more on the financial viability of each option versus looking at what is the *better* solution. Maybe building a product is cheaper and will provide what the customer wants, but by buying it off the shelf the customer can start realizing the benefits earlier, which makes this effort financially viable.

7.3.3 Determine the Approach to Finding a Solution

Often when evaluating the capabilities of an organization relating to new projects, it is discovered that there are alternative ways of achieving the goals. Maybe there needs to be automation. Maybe training existing resources will help. Maybe changing the business processes is the right way to go. This task addresses these issues. It looks at what the alternatives are and which capabilities exist in the organization, assesses the benefit and cost of the different approaches, and recommends which way to go.

One of the techniques previously discussed, SWOT analysis, is useful for this task, but so are others, such as brainstorming and decision analysis.

7.3.4 Define Solution Scope

There must be a link between the business architecture, the evaluation of conceptual solutions, and the project itself. If the business analyst does not clearly understand that link, the project is less likely to hit the target. So how can that linkage be captured? Most of it has probably already been captured in the general project request or project selection process, and it may just be a matter of discovering that information. But often the information needed is inside of the customer's head, which can be a scary place to visit. One of the key problems brought up in the business analyst survey discussed in [Chapter 1](#) is scope creep. The initial definition of the scope is the place to lay

the foundation to avoid scope creep later on. A well-defined scope will clearly show the boundaries of the project. A poorly defined scope will lead to continuous evaluation of what is inside of the project and what is outside.

If the business analyst begins gathering requirements from the stakeholders before the purpose or basic constraints of the project are determined, it is likely that the end product will be something very different from what the customer expected. Some of the key aspects to define to set the stage for the project are

- Business goals and objectives
- Assumptions
- Constraints
- Scope statement
- Impacted organizations

If the business analyst begins gathering requirements from the stakeholders before the purpose or basic constraints of the project are determined, it is likely that the end product will be something very different from what the customer expected.

7.3.4.1 Business Goals and Objectives Here, we are trying to answer the *why* question. Why are we doing this project? What is it that we want to have happen as a result of the project? This could be traditional business goals such as increase profit, reduce cost, and increase revenue, but could also be to meet a new government regulation or respond to competitors. Whatever it is, it is important for the project manager to have a strong understanding of the true business need, and the business analyst can be the best person to assist the project manager in getting that understanding. It is sometimes said that a very good project manager who does not understand the business goals can very effectively deliver the wrong product. So make sure to have clear business goals and objectives for the project. All project objectives should be traceable back to a specific business need. Again, the project scope definition is owned by the project manager, but it is also vital information for the business analyst.

In an ideal world, the business analyst would be involved early in the process and be a part of developing the initial project documents such as the project charter, scope documents, and project plans. If that has not been the case, the business analyst must become familiar with these documents. Not understanding the context of an initiative will often lead to asking the wrong questions or asking the right questions of the wrong people.

7.3.4.2 Assumptions Because much information is unknown at this early stage of the project, the business analyst will need to make some assumptions for planning purposes. This is mainly to make sure that everyone is on the same page. For example, if the customers assume that the product will work globally and the developer assumes

that it will work only in Iowa, there will definitely be problems later on. Typically, in the beginning of a project, there will be a large number of assumptions. As the project life cycle progresses, there should be a discovery that these assumptions are proven right or wrong, resulting (if they were wrong) in a re-planning of the project. Typical assumptions on an IT project may deal with

- Customer availability for sessions and approvals
- Technology availability
- Project priority
- Resource skill sets, availability, and geographical placement

All of these assumptions should be documented.

7.3.4.3 Constraints The project team must understand the limitations under which the project is operating. Constraints limit solution options, and the project team must live within these limitations. Constraints often deal with budget, resources, and schedules. It is important to remember to accept the constraints once they are agreed on by the sponsor and customer. They basically define the box within which the project operates. *Low skills*, for example, would typically not be a good constraint. It is probably something that should be dealt with by hiring a consultant or sending people to training. *Low skills* would more likely be categorized as an issue. More common constraints for an IT project may include

- Existing infrastructure
- Preferred vendor lists
- Schedules (both internal and external)

7.3.4.4 Scope Statement The scope is often high level this early in the cycle, but needs to consider all key deliverables to the customer. The customer should be able to read it and say, “Yes, this is what I want.” This is really a mix of project and product scope. It should contain elements of product scope, such as features included and excluded; however, it should also contain project scope elements such as timing, budget, and release schedules. At this level, the focus is on the boundaries. The concern should be about what is included or excluded, not about the details of any feature. Those details will be captured as the analysis phase gets started.

7.3.4.5 Impacted Organizations This is the start of the communication plan. It tells which organizations to talk to and which ones to actually include on the project team. Although not all stakeholders are equal, they should all be considered. Much of the impact from other organizations can be documented in a context diagram. This shows areas with which the project interfaces and key information going to and from those areas ([Figure 7.5](#)).

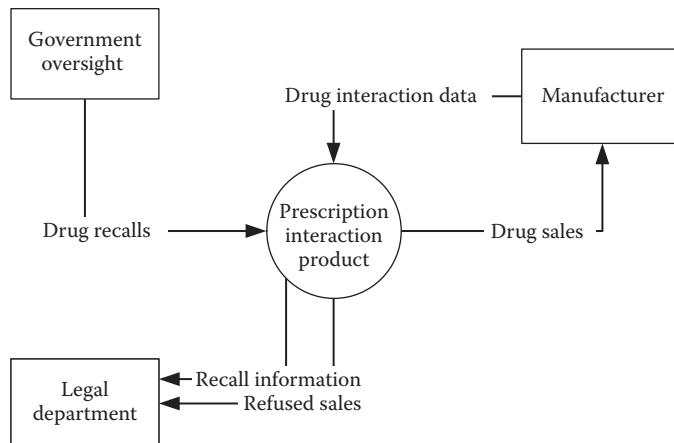


Figure 7.5 Context diagram.

The context diagram, sometimes called the level 0 data flow diagram, consists of three main parts:

1. Scope area: represented by a circle in the middle. This can be a business area such as order processing or accounts payable, but it can also be at a much lower level such as *order prescription drug* or *pay a bill*, the point being that everything inside of the circle is part of the scope.
2. External entities: represented by rectangles. These may be people, organizations, or other systems that will either send information to or receive information from the scope area.
3. Data flow: represented by an arrow. This represents a packet of information going between an external entity and the scope area. These data flows should be key business concepts and are typically easy for the customer to recognize. They can be items such as order, shipping notice, invoice, receipt, or prescription.

Start the drawing of the context diagram by creating the center and labeling it. Then, identify outside organizations, people, and systems with which the product must interface. By asking the customer, “From whom will the system receive information, and what information will it receive?” and “What information will this system need to provide for the outside world, and where will it be sent?”, the business analyst should be able to identify most of these external entities and data flows. It is an activity that should be done together with the customer. The visual representation serves as a catalyst for the customer to think of more interfaces and more data flows. In addition to reviewing this with the customer, it should also be reviewed with the application owners within IT. There are often interfaces needed between systems that customers are either unaware of or just do not think of when they are interviewed. Systems development today is so integrated that very often most of a project’s budget is spent identifying and implementing interfaces, with less being spent on the core functionality the

customer is seeking. This can lead to the customer's questioning why everything takes so long and costs so much. Communicate these interfaces to customers to ensure that they have an accurate understanding of the complexity. Sometimes, customers may decide not to do a project once they realize how severe the impact will be on the rest of the enterprise.

All of this information may have been documented already in the project charter, the statement of work, or a request for service. If those documents do not exist or are not detailed enough, work needs to be done to create or enhance them.

7.3.5 Define the Business Case

Before a project is initiated, there must be a business case made for why the organization should spend money on this project. In its purest form, this would simply be a financial evaluation based on comparing benefits to cost. If the benefits are higher than the cost, the project has value to the organization. However, because most organizations have limited resources and there may be multiple projects that are valuable, a comparison among projects is necessary to evaluate in which one (or ones) the organization should invest. There are many factors impacting this, such as cost to the organization, when the benefits will be realized, and when the cost will occur, as well as other nonfinancial considerations. There may be regulatory reasons as well as market reasons that would cause one project to look better than another. A good business case includes both quantitative (dollars and time) and qualitative (image, gut feel, preferences) information. When making these estimates, both of cost and benefits, the analysts must attach a rating for each factor indicating how comfortable they are with the numbers. Some costs may be known exactly and should be presented that way. Other numbers are good-faith estimates or, in some cases, wild guesses. The numbers must be included either way, but the decision makers deserve to know whether the numbers are solid. This will drive their risk analysis. If all the costs are solid and all the benefits are guesses, then this will be a high-risk project.

Business analysts are not typically responsible for the development of the business case, but they can be participants in the process. By having a good understanding of the business, the business analyst can pinpoint areas of the business that will be impacted by a solution. The business analyst may also be involved with estimating some of the cost factors not only for development of a solution, but also for reengineering the organization. It is a common mistake when doing business cases to be too narrow in the definition of both benefits and costs. Job satisfaction, reduced staff levels, and global presence may be large benefits that are overlooked, just as maintenance, training staff, and turnover represent costs that are often missed.

By having a good understanding of the business, the business analyst can pinpoint areas of the business that will be impacted by a solution.

Clearly identify who will be involved with creating the business case as well as who will approve it. The finance organization, the IT group, and the business owner are all likely to be involved with the estimation of the costs, and most of them will also help estimate the benefits. Before starting the effort to create a business case, the business owner must describe what is required and the format of the case. A business case can be as easy as a benefit and a cost number or as complex as detailed estimates, labor rates, currency exchange rates, and calculations of net present value and internal rate of return. These calculations are normally beyond what the analyst is responsible for.

One of the assumptions that may need to be made while working on the business case is to consider if this will be an in-house development project or if there is commercial off-the-shelf (COTS) software available. A COTS project may be faster and can reduce risk because a working product already exists. However, COTS projects also tend to be underestimated in the area of business impact. Most packages are designed to work within a certain set of business processes. If the customer wants to run the business differently, then there may be a need for extensive customization; if the customer is willing to run the business as the package is envisioning it, then there can be a large organizational impact to update procedures and policies and to train the staff. These are the types of issues that the business analyst can assist with both in identifying as well as quantifying the process of developing the business case.

7.4 Conducting the Initial Risk Assessment

The initial risk assessment is intended to find out if the risk tolerance of the stakeholders is high enough to do this project. All projects are risky; that is part of the definition of a project. A project is unique and temporary, and that leads to high risk. For some organizations, that is good: they live and succeed by being able to overcome risks and deliver products in a high-risk environment. For other organizations, risk can be dangerous. If margins are slim and capabilities are low, then risk can push a business or project over the edge and make it unsuccessful. For the initial risk assessment, there is a set of main questions to be answered by the business:

- How likely is it that we can successfully deliver this product?
- How likely is it that the organization will be able to reap the expected benefits from this project?
- What will the impact be to our business once the product is successfully implemented?
- Is it likely that our business or the marketplace will change by the time this product is implemented?

When doing the risk assessment and recommendation, it is important to look at all these aspects combined. Most organizations are primarily focused on whether the project can be successful, and secondarily if the benefits can be achieved once the project is implemented. It is equally important, sometimes maybe even more important,

that the business evaluates the total impact to the business, both positive and negative, and does a realistic assessment of where the business and market will be by the time the product is implemented. As an example, a business may invest in a new order processing product, streamlining the order processing function and integrating it with inventory control. By the time the product is developed, the business may find that the market has changed and the competitors have gone to an e-commerce model, so the product just developed is obsolete as soon as it is rolled out. Clearly, these trends are not always easy to see or predict, but it is important that the questions be asked, and the initial risk assessment is a good place to ask them.

Risk assessments are done continuously throughout a business cycle and many of the tools and techniques are the same, regardless of when in the cycle they are performed. [Chapter 4](#) has a more detailed examination of the risk management process. At the stage of strategy analysis, the main risk focus will be on impact to business, capabilities, and organizational readiness. As risk is assessed further and further into the project, the more detailed it will be. During the strategy analysis, most risks will need to be negotiated with management. Later in the project, the project manager will own more of them.

7.5 Preparing the Decision Package

There is always someone who will need to make a decision on each idea to determine if it deserves to move on to a full-blown project or if it should be put on the shelf. This may be an executive committee, the business owner, or a product manager. Regardless, the decision maker will need some information to make a good decision. Sometimes, this may be an informal briefing; however, often for large initiatives, there should be a more formal decision package prepared. The business analyst can be a significant player in creating and documenting this package.

The main components of this package are items already discussed in this chapter:

- Feasibility study report
- Business case
- Solution scope
- Initial risk assessment

A good decision package is more than just a summary of these deliverables. It should show that the needs of the stakeholders making the decision are understood, along with how much information they need and the level of product understanding they already have. The decision package should be a clear and succinct summary of the strategy analysis, with recommendations to the decision maker. [Figure 7.6](#) shows an example of a table of contents for a decision package targeting a medium to large initiative in an organization requiring formal approaches.

Although much of the decision package may have been prepared by the business analyst, it would normally be presented to the decision maker by either a project

- | |
|------------------------------|
| 1. Problem description |
| 2. Project description |
| 3. Project background |
| 4. Project benefits |
| 5. Projects costs |
| 6. Risk of doing project |
| 7. Risk of not doing project |
| 8. Alternative solutions |
| 9. Recommendation |

Figure 7.6 Sample table of contents of a decision package.

sponsor or by the project manager. This is a key milestone for any project, and the way the proposal is presented can be just as important as what is presented. It needs to be presented with confidence, using strong executive presentation skills.

7.6 Understanding the Business

Although business analysts must have a good overall understanding of the customer's business, they are not expected to be the subject-matter experts. The business analysts should understand the main drivers of the business, such as business goals and objectives, key competitors, and the customer's market position. They must also have a good understanding of how existing systems are used in the customer's environment.

Some of the most common ways to gain an understanding of the customer's business include

- Selecting business analysts from the business side. Although it is still more common that the business analyst comes from the development side, more and more organizations are now selecting customers with strong IT understanding to be business analysts. That gives the analyst a strong basic business understanding, plus it often provides a network of people who already have a working relationship with the business analyst. And, as will be mentioned many times, relationship building is a key to success for the business analyst.
- Co-locating the analyst with the customer. When working next to the customer, the analyst tends to get more involved with the day-to-day operations of the business as well as building a more personal relationship with the customer.
- Reviewing trade journals, attending conferences, and being part of industry organizations. This adds the benefit of not only understanding the customer's business, but also learning about industry trends and seeing what the competition is doing. It is likely that the business analyst's role will keep expanding more and more into trend analysis and identification of new technologies and solutions.

Whatever methods are used to educate the business analyst, it is important to the success of the requirements-gathering effort that the analyst understands the overall

business picture and can relate how the specific project objectives support or fit in with that business.

In addition, business analysts must view themselves as members of the overall team. Although a business analyst may not have the full knowledge needed, that knowledge often exists within the organization. Knowing who to ask and building a network of people such as project managers, business owners, technical experts, and industry experts will greatly help the business analyst to become successful.

7.7 Business Models

In addition to the context diagram, the SWOT analysis, and the Zachman framework already discussed in this chapter, there are also a number of other models that may be used to help describe and understand the business. Many of these are detailed models from the Zachman framework, and some are documents that exist in most businesses at some levels, but which may not be readily available to the business analyst. The ones to be discussed here are

- Organization charts
- Infrastructure models
- Business location models
- Business events
- Business entity models
- Business process models

7.7.1 *Organization Charts*

Although sometimes overlooked as a model, the organization chart is something that most organizations have and is often easy for the business analyst to access. Traditionally, the organization chart is presented as a number of hierarchically organized boxes, with a name and a title in each of them. However, an organization chart can be much more than that. A good example that is available on the Internet is the organization chart of the United Nations, reproduced in [Appendix C](#) and available at www.un.org/en/aboutun/structure/org_chart.shtml. The value-added portion of this chart is that in addition to illustrating the basic structure of the organization, it also describes what each department or subarea does. Applying this to a business organization, there could be information behind each box stating the vision and mission of the organization, current initiatives, and a description of the functions of the organization. This information is invaluable for the business analyst when working on documenting and understanding the enterprise. If this type of organization chart is not available, it may be good for business analysts to start creating it; however, even if they just have access to the basic, traditional organization chart, that will give them a starting point for more information.

7.7.2 Infrastructure Models

To understand the capabilities of a business, it is often necessary to understand its infrastructure. This can be through a model of worldwide facilities or a model of networks and computers. For the purpose of this discussion, the attention will be on network and computers, but the concepts are valid for any infrastructure.

Figure 7.7 shows the infrastructure model for C.V. Green. Although each global region appears to have some level of internal communication, there is no global network linking all areas. These types of findings are important to the business analyst. Some of the questions that need to be asked include

- Are there plans for a global network?
- Is communication between regions of the world part of the scope of this project?
- Are there legislative or cost issues prohibiting a global network?

Most organizations have some form of documentation supporting their infrastructure model. This will often be held by the IT group, so there may be a need to do some digging around.

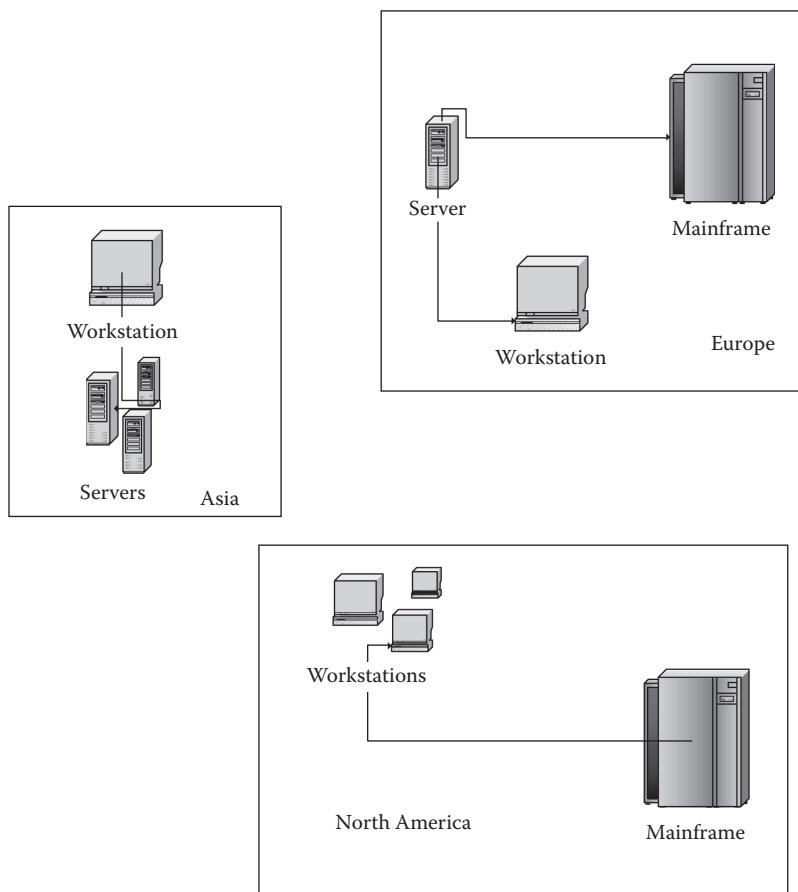


Figure 7.7 Infrastructure model.

This model has a dual purpose. At this time, when trying to understand the as-is situation, it provides a good picture of existing capabilities. It will also be useful for the system analyst, who eventually will be designing a technical solution. Much of the information found in the infrastructure model will convert to being constraints of the solution. Although it is certainly possible that a key corporate project can modify and add on to the existing infrastructure, for most projects the flexibility will be minimal and it will need to live within the existing capabilities. Combining the infrastructure model with the high-level business case can give an indication of how likely it is that significant infrastructure investments will be made.

7.7.3 Business Location Models

There may be overlaps between the organization chart, the infrastructure model, and the business location model. The purpose is a bit different, though: here, the model shows geographic locations in which the business operates. The intention is to find information about local laws, different languages, cultures, and customers, as well as local infrastructure (buildings, utilities, etc.). Why should the business analyst care? In many ways, these things may drive what solutions are possible. If this is a global solution involving three continents and ten countries, it may be difficult to find a single enterprise solution that can be implemented in all locations.

[Figure 7.8](#) shows a business location model for C.V. Green. As seen, any global solution would need to incorporate standards from countries as diverse as France, India, and the United States. There are likely to be differences ranging from the simple, such as power sources and building codes, to the complex, such as trade regulations regarding local content and language issues. The business location model can help in understanding what parts of the business must be regionalized or localized and what parts must be centralized.

7.7.4 Business Events

Much of what a business does is driven by events, sometimes cyclical, sometimes random. Identifying and analyzing these events often leads to a greater understanding of the business and what drives it. The events modeled should be events that require a response from the business and that have a significant impact on the business. Good examples of key business events for C.V. Green are

- Food and Drug Administration (FDA) approval of a new drug
- FDA warning of drug interactions
- Manufacturer warning of drug interactions
- Drug recalls
- Legislative change on drug disclosure

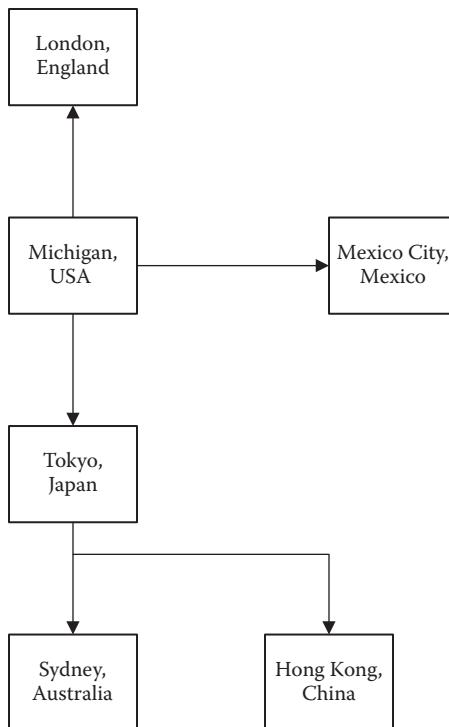


Figure 7.8 C.V. Green business location model.

The identification of events helps when discussing the business with the customer. It helps frame and focus the conversation. Instead of asking what the customer does, the questions should be

- What are key events in your business?
- How do you respond to these events?

Event models work well together with creating business scenarios. Rather than abstractly reviewing with the customer what actions are taken when the FDA issues a drug interaction warning, pick a specific case and walk that through to completion. For example, “the FDA just identified a harmful drug interaction between drug A from manufacturer X and drug B from manufacturer Z.” Building this scenario will give a better understanding of the business process as well as help identify key decision points in the process. Questions such as “Does it matter if the drugs targeted by the warning come from the same company or from two different ones?” should be asked. These types of conversations tend to be less ambiguous and more valuable for both the customer and the business analyst.

One advantage of events modeling is that the customer seems to relate to it. Most customers look at their business in the context of responding to events, normally from the outside; however, the events could also be based on time, such as *month end* or *tax*

deadline, so this form of modeling is likely to engage the customer in the process. Be aware that not everything the customer does is in response to an event. Things such as *price increase* and *take inventory* are both actions that may be taken without being initiated by any specific event.

7.7.5 Business Entity Models

Business entity models can be done at multiple levels (Figure 7.9). For example, they can be looked at within a business to see how departments are relating to each other, or they can be done between companies to understand the organizational interfaces. Viewed from an internal focus, as in the example of C.V. Green, the business entity model can show with what other areas of the business the pharmacists deal. Do they need to get information from accounting? How is inventory determined? Are they interfacing with legal? Looking at it from the external viewpoint, the focus is on business impacts on the outside world, and outside impacts on the business. Business entities could be *customer*, *vendor*, *FDA*, or *drug manufacturer*.

An example of a business entity model is shown in Figure 7.9. The arrows going back and forth between the business entities and the organization show the main flow of information or products. This model is helpful in gaining a big-picture understanding of what outside organizations provide products, services, or information. That does not mean that these organizations will all be interviewed, but their impacts will be considered. A business analyst starting to gather requirements for the Prescription

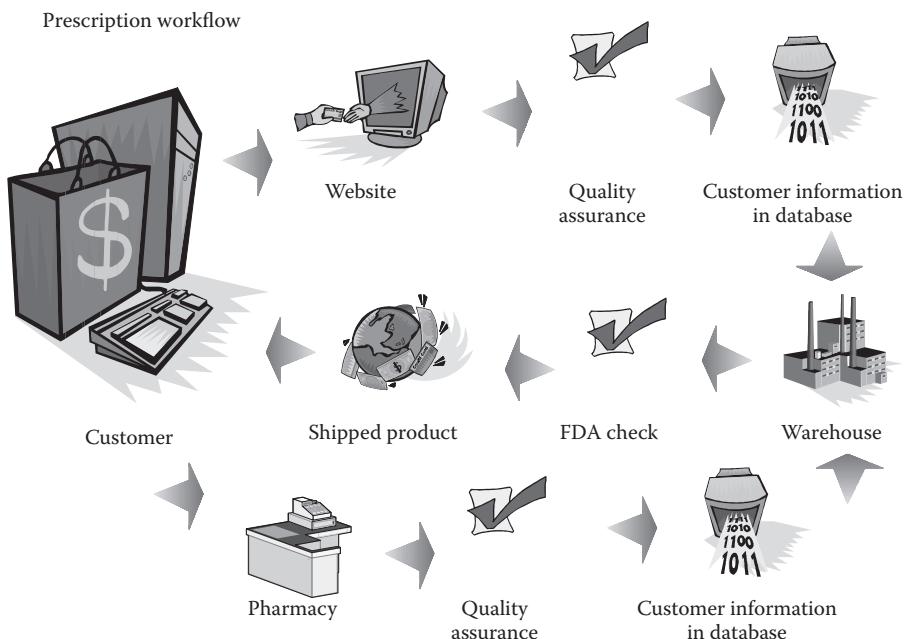


Figure 7.9 Business entity model.

Interaction Project may not understand that the pharmacy gets some of its information on drug interaction problems directly from lawsuits and that any good solution to the problem must include capturing lawsuit information as early as possible. This type of information can be caught by reviewing the business entity model before planning the requirements-gathering efforts.

7.7.6 Business Process Models

Perhaps the most common set of models used to describe the business are process models. There are many types of process models, including workflow, flowchart, and swim lanes, just to name a few. Some of these are discussed in [Chapter 8](#) as a part of requirements documentation and modeling. In the context of this unit, they are being reviewed at a higher level, as an approach to understanding the business. An example of a business process model is shown in [Figure 7.10](#).

Looking at the model, it is easy to see that there are many steps in identifying interacting drugs and some of them reside outside of C.V. Green. This will help

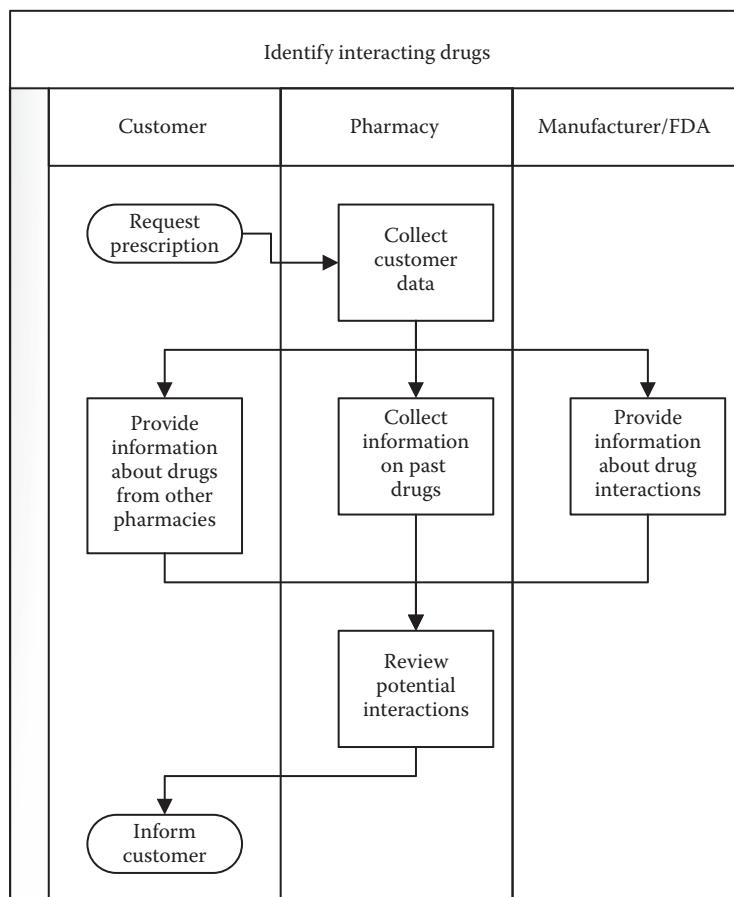


Figure 7.10 Business process model.

determine the scope of the undertaking and how feasible the whole effort is. In this case, there will need to be considerable cooperation with businesses and organizations outside the immediate organization, with which there is usually little communication. Therefore, if the business wants to take on this endeavor, it must be prepared for a large undertaking.

Other types of process models include hierarchical process structures, input-process–output (IPO) models, and business use case models. An example of a hierarchical model is a functional decomposition diagram (FDD). IPO models include data flow diagrams (DFDs). FDDs and DFDs are covered in [Chapter 8](#). Business use cases are similar to the use cases covered in [Chapter 8](#), but take a business view rather than a systems view. Business use cases take an external view of the business. What does the outside world want to do with this business? Using FDDs and DFDs takes an internal view, looking at what the business does, not what the customer wants it to do. This difference in approach should be considered when selecting which technique to use. Will this project primarily focus on an external customer view of the business, or is it being driven by internal needs?

Most projects will deal with impacting business processes in some way, so there is always a need to understand those processes. Make sure that the business is heavily involved with creating these models. If they are IT related, then they tend to put too much emphasis on what the technology does, and while that is important, it is not where the focus should be at this time.

7.8 Summary

The business analyst typically works on projects that support the enterprise in some way. Understanding the business drivers behind the project and ensuring that the business analyst has a solid understanding of the customer's business will greatly enhance the chance to successfully elicit and document the right requirements. Hopefully much of this work has already been completed by the business; but if not, it should be done at the beginning of the analysis phase. On a small project or on an enhancement project, little effort may be spent in this area; however, for a large, enterprise-wide initiative, this area deserves serious attention up front.

Activity

Review the Swede-Mart case study in [Chapter 10](#) and fill out the following charter for the project:

PROJECT CHARTER

Project name:

Customer:

Project manager:

Project sponsor:

Business need/issue:

Project justification:

Critical success factors:

Key product deliverables and milestones:

Organizational assumptions:

Organizational constraints:



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

REQUIREMENTS ANALYSIS AND DESIGN DEFINITION

To succeed in business it is necessary to make others see things as you see them.

John H. Patterson

This chapter reviews the best, or at least the most commonly used, practices for modeling and documenting requirements. Each of these techniques warrants its own book(s). However, the purpose of this book is to present them at an introductory level, giving the business analyst enough exposure to the techniques to be able to participate in a requirements-gathering or review session where these techniques might be used. It is also intended to help in selecting which techniques to use. The analyst leading these activities should further develop his or her skill in the techniques used for the project. Although modeling can be of great value when gathering requirements, if the business analyst struggles with it, the customer will soon lose interest as well.

Knowing how to create a structure that enhances comprehension and makes the analysis phase easier can be useful and is actually critical for success on large and complex undertakings. There are many approaches to categorizing requirements, and some of the more common ones are explored in this chapter. The International Institute of Business Analysis™ (IIBA™) approach to classification is reviewed, along with other equally good approaches. It is not the intent in this chapter to promote one specific classification system; rather, it is my view that different projects and different business situations need different classification approaches, and the reasoning for this will be explained in this chapter as well.

8.1 Objectives

- Realize the value of modeling as a part of documenting system requirements.
- Recognize and read each model type.
- Gain an understanding of which area of the system each technique is modeling.
- Understand the strengths and weaknesses of each technique.
- See the applicability of each technique to the project being worked on.
- Practice modeling.
- Identify the need for categorizing requirements.
- Determine to what level of detail a requirements-gathering effort should go.

- Review different categorization schemes.
- Review situations where each scheme of categorization may work.
- Determine roles and responsibilities involved with different categories.

8.2 Overview

Sometimes, starting to capture and document requirements seems like a daunting task. It is difficult to know where to begin and into what level of detail to get, and it is almost impossible to know when a complete set of requirements has been captured. A well-defined classification system, or taxonomy, can help with these issues. It is similar to going to an automotive dealer to buy a car. It helps to organize the discussion into categories. For buying a car, those categories might be interior, exterior, and performance related. For a system, it will be different categories, but the purpose is the same: to enhance communication.

It is almost impossible to know when a complete set of requirements has been captured. A well-defined classification system, or taxonomy, can help.

Although the main purpose of this book is to discuss requirements-gathering techniques, to do that without looking at how to best document those requirements would be to miss a critical component of the requirements process. Requirements models and text documents are used not only as a repository for information, but they are also key components of the communication process. Combining the use of models and text to reach a total understanding of the customer's needs is a great best practice in the field of business analysis. Text without pictures tends to be difficult to read and difficult to organize. Models without text can be fluffy and not contain enough information to understand the detailed requirements of what is being developed. Best practice is to do both. Set the context, the overview of the business and system, with models and pictures showing the overall scope and flow of what is being worked on, and then use the model as a communication tool to capture the detailed information in accompanying text documents.

[Figure 8.1](#) shows models and text working together. This technique is especially effective when working with the customer in a requirements session. The customer stays focused by exploring one piece of the model at a time and delving into the details of what the business requirements are for that part of the business. Without the model to keep both the customer and the business analyst focused, there is a tendency to randomly move from one area of the system to the next, and then back again without having a logical direction.

Without the model to keep both the customer and the business analyst focused, there is a tendency to randomly move from one area of the system to the next, and then back again without having a logical direction.

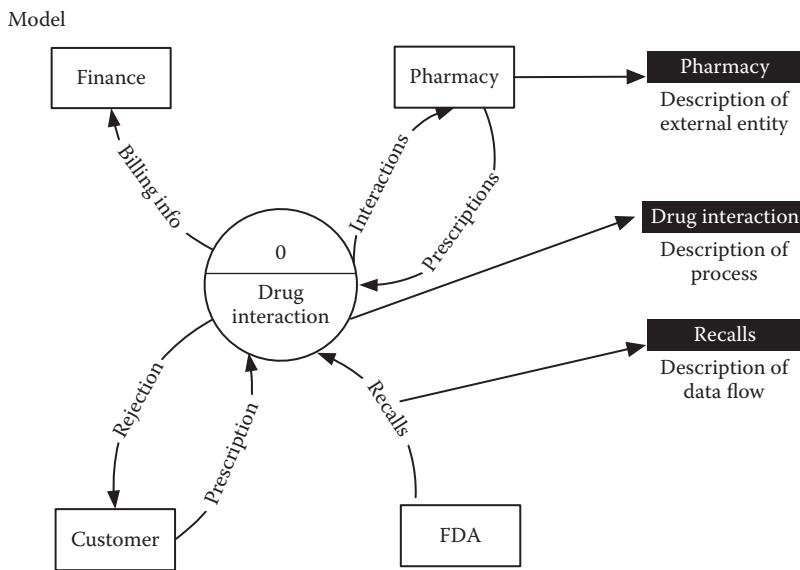


Figure 8.1 Models and text.

The two topics, models and text documentation, have been combined into one chapter, because the outcomes will be used together to understand the business. However, the skill set needed for each of these is somewhat different. Good modelers must have a mind suited for abstractions, pictures, and simplification. They must also understand what level of detail is appropriate for a user versus a developer. Their style of communication must be facilitative, with the ability to develop and review the models with a group of people.

Technical writers need a different skill set. They must be very detail oriented, thorough, and precise to make sure that sufficient clarity has been accomplished. It is often difficult to find a single person who can successfully perform both tasks. In many cases, it is best to team up a couple of people to perform these tasks, with the big-picture person focusing on the models and the detail-oriented person making sure that the models have real information behind them in the form of text documents.

Although there are many modeling techniques, and many ways of grouping them, for the purposes of this chapter there will be two major groups:

1. Traditional models
2. Unified Modeling Language (UML, or object-oriented) models

Traditional models are rooted in structured analysis and design, and UML models in object-oriented concepts. Some modeling techniques can be used in both environments, such as use cases and activity diagrams. In this book, the activity diagrams are covered as a part of UML; however, there are other notation techniques for the activity diagrams. Those notations would be used in a traditional development environment. The concept of activity diagrams is the same in both cases. There is no attempt

in this book to make a case for a specific technique. Rather, by looking at different approaches, the recommendation is for each organization to evaluate which ones best fit the developers, customers, and types of development efforts that are being managed. For example, research and development needs different techniques than legacy systems, upgrades, etc. A project to enhance an organization's existing accounts payable system, developed 30 years ago on a mainframe platform and integrated with other legacy systems, should normally use structured analysis and traditional modeling techniques. Since those were more than likely the techniques used originally when the system was developed, those techniques will likely fit the best into the existing architecture. In addition, there may be existing models and documentation from the original development that may be reused, saving both time and effort. On the other hand, if this project is an organization's first venture into e-commerce and web development, then it would make much more sense to look at object-oriented approaches because the tools that will eventually be used to develop the system more than likely will have that orientation. Because most projects probably fall somewhere in between those two extremes, the decision may not always be clear; however, the goal should be to look at the development environment for the system.

Although it is often acceptable, and sometimes even desirable, to mix and match between the approaches, the awareness that some techniques within one family of diagrams overlap with techniques from the other area is essential. Keep in mind also that, as with most efforts, there should be an attempt to minimize the duplication of effort when modeling. The concept of *one fact in one place* is a good goal to have. In the end, the most frequent determinants of which technique to use is a combination of the experience of the people involved and the tools in which the organization has already invested. If the organization has invested hundreds of thousands of dollars in a tool suite, it should be used. These tool suites, whether they are Oracle™, Rational™ (IBM), or a mix of vendor products, are intended to help with a significant portion of the development cycle. The training, the tool itself, and the experience built up with a tool are invaluable for an organization, and there must be a very strong and substantial reason not to use those tools for any projects within the organization.

In the end, the most frequent determinants of which technique to use is a combination of the experience of the people involved and the tools that the organization has already invested in.

8.3 IIBA Tasks

The IIBA has identified six tasks that fall under requirements analysis:

1. Specify and model requirements
2. Verify requirements
3. Validate requirements

4. Define requirements architecture
5. Define design options
6. Analyze potential value and recommend solutions

8.3.1 Specify and Model Requirements

When sophisticated business computer systems became more prevalent, the need for an organized approach to develop those systems became a higher priority. Some people referred to it as *structured design*; another popular phrase was *information engineering*. The basic concepts of this approach were to model data and processes separately, often in a functional decomposition diagram (FDD) and an entity relationship diagram (ERD), and then model the interaction between data and process in a data flow diagram (DFD). These modeling techniques are still used today, sometimes combined with use cases or other approaches to show process requirements. **Table 8.1** shows the structured analysis techniques discussed in this section along with the purpose of each, as well as possible alternative techniques that would provide a similar view and purpose.

8.3.1.1 Process Models Functional decomposition is one of the most common techniques for documenting requirements. The diagram starts at a high level (business area) and then gradually decomposes into key functional areas and processes within those areas. This is a good way to pinpoint scope for a project, as well as identify which stakeholders will be able to provide further information and requirements. It also tends to view the business the way the customer views it, making it a fairly customer-friendly tool.

As shown in **Figure 8.2**, the business area (at the top of the diagram) could be the whole enterprise or as small as a single process (*Enter patient information*). That is the starting point for a decomposition of the process into a higher degree of detail. How far down into the details should the decomposition go? The lowest level, referred to as the basic business process, should identify what the business needs to do without specifying how it is being done. If the lowest level becomes a how-to instruction or if it contains physical implementations, then there is too much detail, although it is

Table 8.1 Comparison of Analysis Techniques

TECHNIQUE	MAIN PURPOSE	ALTERNATIVE TECHNIQUES
Functional decomposition diagram	Describes business processes and functions from an internal view	Use case IDEFO
Entity relationship diagram	Describes information needed for the business and the business rules regarding that information	Workflow diagram Class diagram
Data flow diagram	Shows the flow of information in and out of business processes, and how the customer views and groups the data	Input–process–output models State diagrams

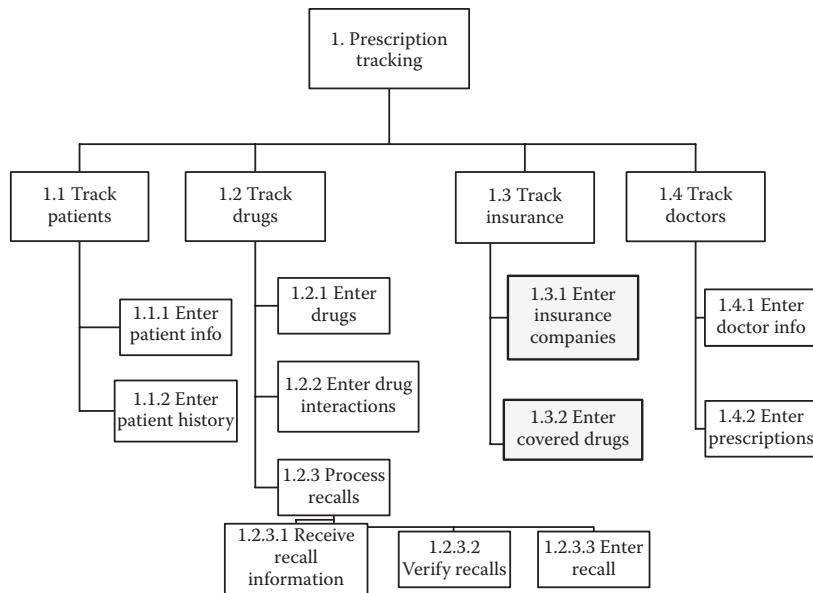


Figure 8.2 Functional decomposition diagram.

often difficult to draw that exact line, and there is usually a certain amount of intuition involved with that decision. The lowest level should be a detail of the *what* without going into the *how*.

Figure 8.2 shows a step-by-step approach to creating the FDD:

1. Identify the focus area; in this case, it is *Prescription tracking*. That is the scope of our project. Naturally, different people will look at those two words and read something different into them. So for every box in the FDD, there must also be, in a supporting document, a description that clarifies what is meant by the text in the box, in this case *Prescription tracking*. In some cases, this has already been documented in the project scope statement, and if so, a reference to that document is enough. In other cases, the analyst may have to create a description and make sure that the customer agrees with it. Modeling can be of great value, but only if the analyst and the customer are modeling the same things.
2. Ask the customers what the main business functions or processes are within the area of *Prescription tracking*. In this case, the customer came up with *Track patients*, *Track drugs*, *Track insurance*, and *Track doctors*. In an FDD, there is no concern for sequence, so the processes (on the same level) can be shown in any order. Complete each level of the decomposition before moving on to the next level. Ask the customer if there is anything else that needs to be done within *Prescription tracking* that is not covered by the four high-level processes already identified. One of the rules of the FDD is that the sum of all the lower-level processes must have the same scope as the level just above them, only with

more detail added. All activities from the *parent* level must be reflected somewhere on the *child* level.

3. Now review each one of the four processes from step 2 to determine what the detail processes are below them. For the process *Track drugs*, the customer identifies *Enter drugs*, *Enter interactions*, and *Process recalls*. Again, there must be a balancing at the borders, so that the process of *Track drugs* is fully described by the three detailed processes below it and all of the detailed processes are necessary to describe that process.
4. Next, review all the detailed processes and see if any of them needs further decomposition. In the example, the process *Process recalls* can be decomposed into *Receive recall information*, *Verify recalls*, and *Enter recalls*. None of the other processes needs further decomposition. There is no need to have all processes broken down into the same number of levels. Rather, let the business decide how many levels each process should have. In some cases, the customer may want to go into too much detail. As an example, look at the process “Enter patient info.” That could be broken down into “Bring up patient screen,” “Enter name and address,” “Enter phone number,” and “Enter preferred pharmacy.” This is too much detail and should be avoided by applying the following tests:
 - a. Is the process describing the use of a tool, such as “Bring up patient screen”? If so, it does not belong here. Remember, the process should describe *what* the business does, not *how*.
 - b. Are the processes mostly step-by-step instructions on how to do the higher-level process? “Enter name and address,” “Enter phone number,” and “Enter preferred pharmacy” fall into this category.
- There is no absolute test to see if there is enough detail. It is good to remember the purpose of the diagram: it is a tool to help communication. If the customer insists on more detail, then provide more detail. But if the diagram gets too cumbersome to read, then consider staying at a higher level.
5. Determine if any of the boxes are outside the scope of the project. In the example, the processes of “Enter insurance companies” and “Enter covered drugs” are determined to be out of scope, covered by some other project, so they are marked with a shaded background. This is one of the great advantages of FDDs: they provide opportunities to graphically depict what is inside the scope and what is outside.
6. Each process in the FDD must then be documented using a standard template, such as the one shown in [Figure 8.3](#).

It is in this template that most of the real information needed by the developers is captured. Both functional and nonfunctional requirements can be documented here. The role of the diagram is mainly to give structure and context, while the template gives the detailed information. In the initial stages of documentation, this template can be high level and focus on the business requirements; however, as the

Process description	
Process name: 1.2.3.1 Receive recall information	Documented by: Jane Analyst
Documentation date: January 10, 2007	Process owner: John Customer
Process description: This process receives drug recall information from the drug manufacturers or from government agencies and starts a formal tracking of each recall request.	
Process steps: 1. Receive recall physically, electronically, or through phone call 2. Capture all relevant information 3. Enter information into tracking system	
Business rules: All information must be entered into tracking system within 12 hours Any emergency requests must be escalated to management within 1 hour	
Is triggered by (other process or event): Government recall Manufacturer recall	
Is trigger for (other process of event): 1.2.3.2 Verify recall Emergency escalation process	
Comments High visibility process	

Figure 8.3 FDD process description.

requirements-gathering effort progresses, this document should eventually show systems requirements as well.

Strengths of FDD:

- It is intuitive for the customer: Looks at the business the way customers do.
- It gives traceability between detailed business process and the business functions where they are performed.
- It helps discovery of duplicate and overlapping activities.

Weaknesses of FDD:

- It takes an internal view of the business or system (what the organization is doing, not what the external customer wants to do).
- It is difficult to decide how much detail to get into.

Best practices:

- When working with executives, take a top-down approach, which tends to mirror the way they view the business.
- When working with end users, do a bottom-up approach, which will describe the level they work at, and build the structure afterward.

8.3.1.2 Data Models The FDD shows the business processes within an organization. Because, for the most part, the business analyst is defining what an information system will do for a customer, there is also a need to define what information (or data) that system will actually need. This is not intended to be a definition of what the computer needs, but rather what data are important to the customer and what the business rules are guiding the use of that data.

The most widely used technique to show data needs within a business is the ERD. One of the pioneers in the development and standardization of the ERD was Dr. Peter Chen in the 1970s. It should be noted that creating ERDs is a large and complicated process, which is only covered at a high level in this book. There are different ways to document them; many toolmakers use their own drawing standards, but they are all very similar. The basic components of an ERD are shown in [Figure 8.4](#), with some alternative notations shown in [Figure 8.5](#).

The key information captured in an ERD includes

- Entities—things, people, or organizations about which the business needs information. Typical examples are customers, orders, and employees.
- Attributes—the information that needs to be kept for each of the entities. Examples are name, address, description, and quantity.
- Relationships—the business rules that govern the relationships between entities. Examples include “Each customer may place orders” and “Each order must be placed by one customer.”

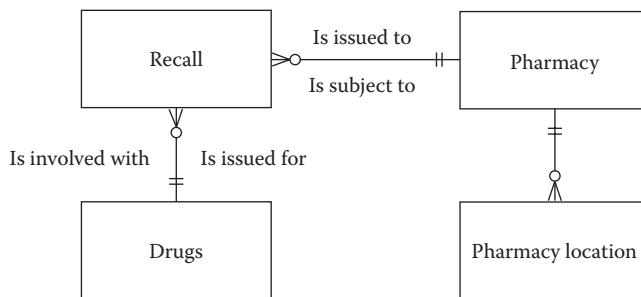


Figure 8.4 Sample ERD.

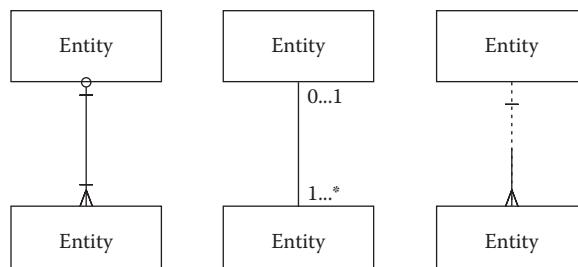


Figure 8.5 Alternative notations for cardinality.

Similar to the FDD example above, there is a step-by-step approach to creating and verifying an ERD:

1. Identify entities. Start by identifying the main concepts that are important to the customer's business. Listen to the customer's description of the business and then select the concepts that sound like items about which the business may need to keep information. Key nouns are often candidates for entities. The customer description of the business is, "Within prescription tracking, we keep track of which patients have been sold what drugs and what doctor prescribed them. That information must be transferable to all pharmacies within our network. We are also responsible to track recalls and drug interactions, which may be initiated by the manufacturers or the government." When evaluating that statement, the analyst initially found the eight underlined concepts as potential entities. Each one of them should be evaluated against the following criteria:
 - a. Is the potential entity part of our scope?
 - b. Do we need to keep any information about the potential entity?
 If both of those answers are positive, then add it to the list of entities.
2. Identify attributes. Next determine what information the business needs to track about each entity. For the patient, the customer tracks name, address, and phone number. Those are the attributes of the patient entity. It is worth remembering here that the only attributes identified should be the ones that are important for the scope of this project. There may be other attributes for this entity that are used by other projects, but those should not be included here.
3. Identify relationships. The third step is to identify the relationships and business rules between the entities. This helps answer questions such as "Who issues a recall?" and "How many manufacturers can there be of a drug?" The answers will identify the relationships. Each relationship between two entities is bidirectional and must be described by two business rules, one in each direction. [Figure 8.6](#) provides more details of the relationship between *Recalls* and *Drugs* from [Figure 8.4](#).
4. Determine cardinality. There are two symbols at the end of each line that describe what is called *cardinality*. The one furthest away from the entity identifies whether the relationship is mandatory or optional (marked with a straight line across the relationship for mandatory and a circle for optional). This is called the *minimum cardinality*. The other symbol, the one closest to the entity, describes the maximum number of occurrences of this entity that can be related to an occurrence of the other entity in the relationship (a maximum

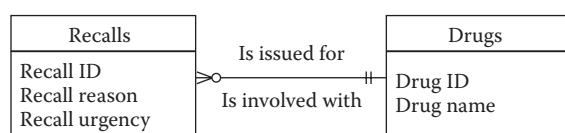


Figure 8.6 Recall/drug relationship.

of one is marked with a cross line, and a maximum of many is marked with a crowfoot symbol). This is called *maximum cardinality*. Sounds complicated? It can be, and it does take some time to get used to reading data models.

Review [Figure 8.6](#) again, starting on the left side. The first business rule reads “Each recall is issued for a minimum of one drug and for a maximum of one drug.” Always start the statement with *Each*. A business rule is always read clockwise so the statement for this rule is on top of the line (is issued for). Just to the left of the drug entity are the two cardinality symbols, in this case two cross lines. The first cross line indicates that this is a mandatory relationship and the second cross line indicates that there can only be a maximum of one drug for a recall. Read the second business rule: “Each drug is involved with a minimum of zero recalls and a maximum of many recalls.” Again the statement starts with *Each*. Because the reading direction is clockwise, the relationship words are under the line “Is involved with.” The cardinality symbols are to the right of the recall entity. The first one is a circle, indicating that this is an optional relationship (not all drugs will be subject to recalls). The second one is a crowfoot, indicating that each drug may be subject to many recalls.

Some alternative notations for cardinality are shown in [Figure 8.5](#). If a tool is used to create the ERD, it is best to use the same technique used by the tool. Stay consistent! ERDs can be confusing enough for the people reviewing them, without changing conventions.

5. Document. The final step is to create the documentation for each entity, attribute, and relationship. An example of entity documentation can be seen in [Figure 8.7](#). Sometimes, it is easier to review this text documentation with the customers and leave the actual ERD on the sideline. ERDs, although critical for a successful software project, are not always easy for the customers to understand.

Entity: Customer	
Owned by: Sales department	Documented by: Jane Analyst
Date: June 1, 2007	Reviewed by: Joe BusOwner
Entity description: A customer is defined as anyone who has purchased a product or placed a prescription with the pharmacy in the past three years.	
Attributes: Name Address Phone number	
Relationships: Each customer may have many prescriptions Each customer may have many bills	
Comments/issues: Common entity used by many systems. Any changes must be coordinated through the sales department.	

Figure 8.7 Entity documentation.

Over the years, data models have evolved into three main categories:

- Conceptual data models—primarily focused on communicating with the customer. These often show only key entities, attributes, and relationships. Data models can get confusing, and the conceptual data model attempts to stay at a level where the customer is comfortable.
- Logical data models—focused on giving the developers the information necessary for development. They need to be fully defined and normalized. Normalization is a data modeling concept that can get complex; however, for this book, it means ensuring that one fact is in only one place.
- Physical data models—look at the technology and how the data will be used by the customer to determine the actual physical implementation.

Both the conceptual and the logical data models fall in the domain of the business analyst. Often the organization will have data architects that assist with this, but the overall responsibility is the business analyst's. The physical data model is for the developers. It should be noted that the conceptual data model and the logical data model can be the same. If your customer can understand the same level of detail and documentation that the developers need, it is OK to use the logical data model as the conceptual data model.

Along with the data model is the data dictionary. Each attribute (or data element) must be defined. Some of the key parts of that definition include

- Name.
- Aliases. It is very common, especially in large businesses, for different parts of the business to have different names for the same data. One business unit may call it *vendor* while it is called *supplier* by a different unit.
- Description, or definition.
- Format. This can be a number of formats such as numeric, text, date.
- Values. What are the valid values or range of values?

Although it is difficult to standardize data definitions across an enterprise, it is important to standardize the definitions as much as possible across each business unit. Inconsistent definition will lead to miscommunication and poor requirements definitions. Many organizations leave the data definitions and much of the data dictionary to the system analyst, rather than the business analyst, which can cause problems. The system analysts will often make decisions based on what they have seen in the past, which may not reflect how this specific customer is running and defining the business.

Strengths of ERDs:

- They provide one place for all data needs for a system.
- They serve as repository for future reuse.
- They are easy for developers to use.
- They ensure integrity of information.

Weaknesses of ERDs:

- They can be difficult for the customer to comprehend.
- They can be difficult to administer for large functions.
- Different parts of the business have different definitions of data.

Best practices:

- Review existing business descriptions and models to identify potential entities.
- Review forms and screens to find potential attributes and relationships.
- Use the customer's language to describe all components.

8.3.1.3 Data Flow Diagrams In the traditional approach, the process model shows what the business does, and the data model identifies what information the business needs. The purpose of the DFD is to tie these two things together. The DFD ties the processes identified in the FDD with the information that is used and created by those processes. DFDs are done in levels, with each level tying into a level of the FDD, as shown in [Figure 8.8](#).

The top level of the DFD, often called *level 0* or the *context level*, is primarily a scope definition tool. It shows the boundaries between the project and the outside world. An example of a context-level diagram is shown in [Chapter 7, Figure 7.5](#), where the scope of the project and its ties to the rest of the enterprise are explored. Although often owned and developed by the project manager, the context diagram is a key document for the business analyst as the requirements-gathering process starts.

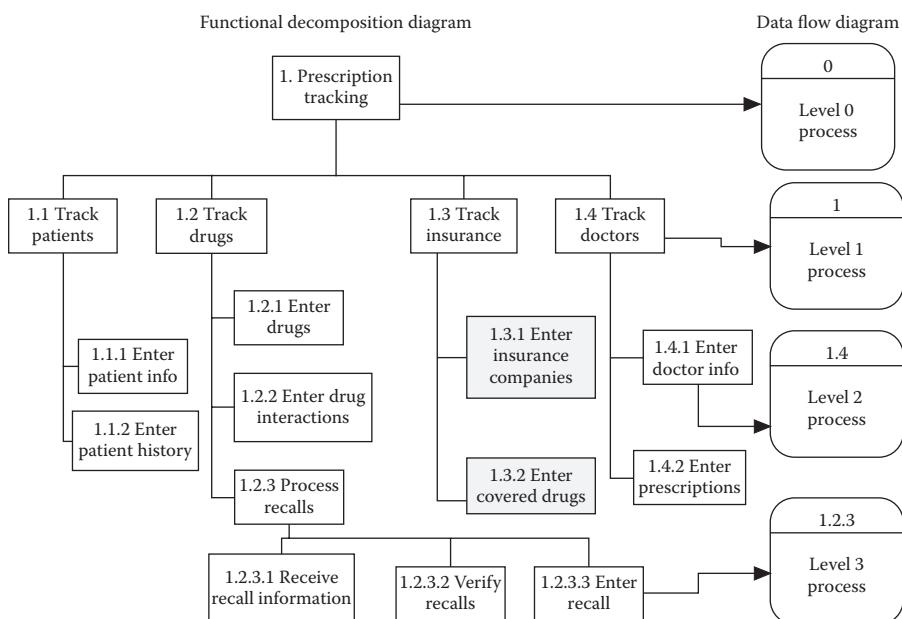


Figure 8.8 FDD and corresponding DFD levels.

The levels below the context diagram become increasingly focused on the internal dealings of the business system as the processes and functions are decomposed, mirroring the decomposition of the FDD. [Figure 8.9](#) shows an example of a level 1 DFD.

The key components of the DFD are

- Process—the activity or function done by the business
- External entity—the people, systems, and organizations outside of this business area that will provide and/or receive data from the processes
- Data flow—the data flowing between the processes and the external agents
- Data store—data that needs to be kept by the business for some time until some other process needs it

The three processes shown in [Figure 8.9](#), *Track Patients*, *Track Drugs*, and *Track Doctors*, are the same three processes shown at level 1 of the FDD in [Figure 8.2](#). This is how the two models interact: each level of the FDD corresponds to another level in the DFD. In the example shown, there are four external entities: *Patient*, *Doctor*, *FDA/MFR*, and *Physician tracking system*. Those are the people, organizations, or other systems that are outside of the control of the Prescription Tracking System, but there is a need to interface with them. Any data flows going between those external entities and the processes being evaluated must be documented. A data flow is a concept that is real in the customer world. It does not always translate neatly to the data model, but rather

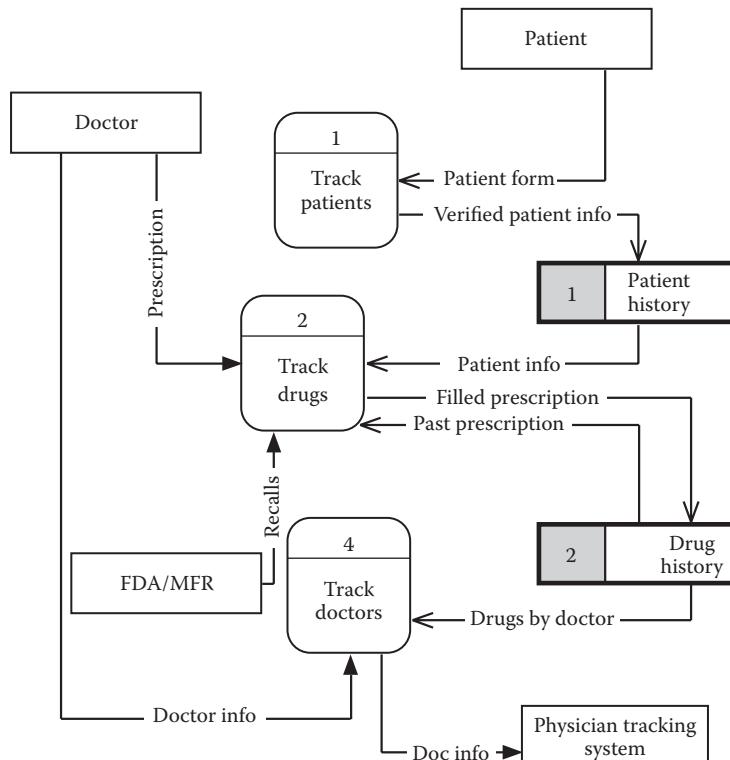


Figure 8.9 DFD level 1.

it tends to be a combination of data from many entities. Typical examples of data flows are *Invoice*, *Bill*, and *Registration*. Later in the project, mapping between the data flows and the data model will be done to determine where the information will be stored. The last component is the data store. A data store recognizes that the business may need to keep information internally for different purposes and different time frames. Again, it is not a piece of the data model, but more a view of the data the business works with. It does not actually have to be stored in a physical database; it can be just a stack of papers. Look at the example of a telephone bill. It can be unpaid and it can be paid. If the process being defined used the unpaid bill as an input and the paid bill as an output, then the model would show those as two different data stores. In [Figure 8.9](#), the model shows *Patient history* and *Drug history* as the two data stores of importance.

A data store recognizes that the business may need to keep information internally for different purposes and different time frames.

To create the level 2 DFD, each process on the level 1 DFD could, if needed, be decomposed to its detail processes. In this case, that would result in three level 2 DFDs, one per process. In [Figure 8.10](#), the process of *Track Drugs* has been decomposed to the next level.

Again, the processes at the next level should be taken from the next level of the FDD. That will ensure those two models remaining synchronized. One important concept of DFDs is that every level of the model must have the same borders as the process above it. This means that any input/output to one level of the model must

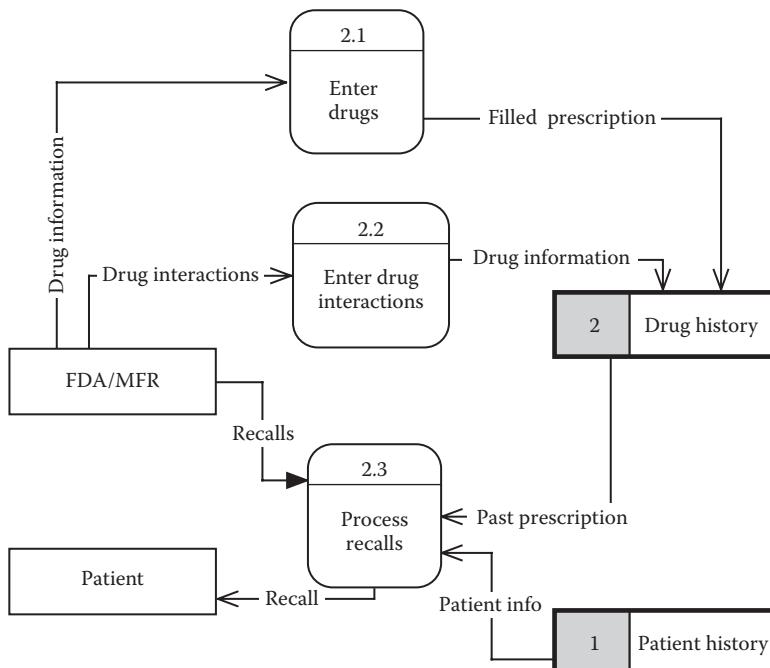


Figure 8.10 DFD level 2.

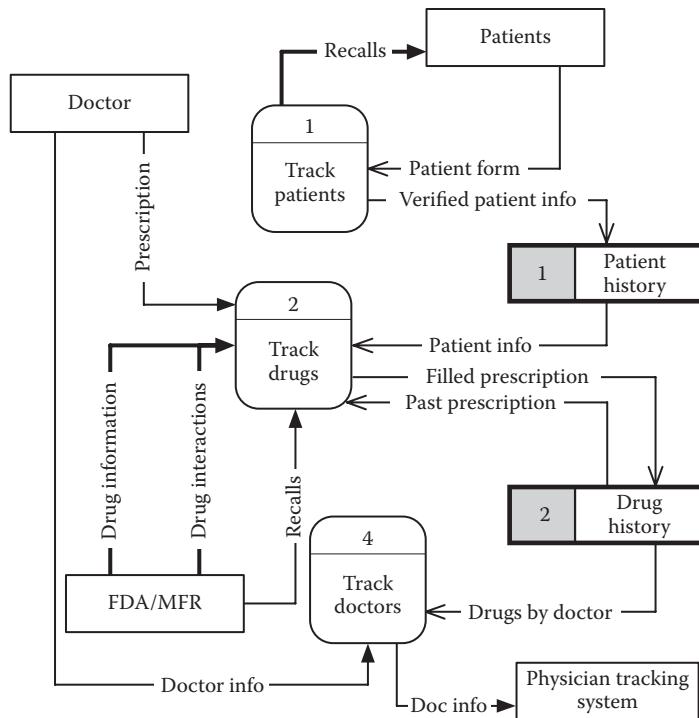


Figure 8.11 DFD level 1 corrected.

correspond with the process above it. Compare [Figure 8.10](#) with the process *Track drugs* in [Figure 8.9](#). Are there any differences at the borders? [Figure 8.10](#) shows *Recall Information* going to the patient and *Drug Information* coming from FDA/MFR. Those data flows are not shown at the parent level in [Figure 8.9](#). That is an error, and it must be corrected. If an advanced tool was used to create the DFD, it would prevent this from happening. But if these diagrams were drawn with simpler (and cheaper) drawing tools, these borders must be controlled manually. [Figure 8.11](#) shows a corrected level 1 DFD. The problem can be in either (or both) of the level 1 or level 2 diagrams. The analyst must get the correct information from the customer.

8.3.1.4 UML Family UML is defined by the Object Management Group and is a nonproprietary language for object modeling, even though many organizations today use it beyond the boundaries of object-oriented (OO) development. In the context of this book, it is viewed as a software modeling approach; however, it can be used to model business and systems engineering. UML contains a lot of modeling diagrams, and there are many ways to organize them. The method selected here helps to take a hierarchical view and uses three major concepts:

1. Structure: Models such as class diagrams and object diagrams fall into this category.
2. Behavior: Activity diagrams, state diagrams, and use cases are examples of this.
3. Interaction: Examples include collaboration diagrams and sequence diagrams.

The UML evolved in the early 1990s with inputs from many sources, culminating in the efforts by James Rumbaugh, Grady Booch, and Ivar Jacobson, commonly known in the industry as “the three amigos.” Their UML 1.0 draft was submitted in 1997. It has gone through many revisions, and UML 2.0 is the current version.

UML is a set of recommendations and guidelines, most of which are optional and need to be customized by the organization developing the system. It does tend to be used in a way that makes it appear closer to the developer (system-centric) than to the customer (business-centric), so organizations sometimes struggle with using it in the very early stages of a project when it may not yet have been decided whether a new system is needed.

When should UML be used rather than a more traditional approach? It tends to be primarily driven by the development environment. If the organization’s development methodology is object oriented, then UML would normally be the technique of choice. If the development environment is traditional, then the modeling techniques would likely be the traditional ones reviewed in the early part of this chapter.

When should UML be used rather than a more traditional approach? It tends to be primarily driven by the development environment.

Structure Diagrams—Structure diagrams focus on what needs to be included in the system being modeled. They include models for data, hardware, program modules, files, etc. One of the more common diagrams that a business analyst may be exposed to is the class diagram.

Classes basically define components within a system, such as customers, orders, parts, and other key components. Classes can also be used to define hardware- and system-related items; however, in general, the business analyst must be careful in deciding what level of detail to use for these diagrams. In their simplest form, they can be very usable, and they can look a lot like a data model, but they also include processes, called methods. So a class, such as customer, can have attributes such as customer name and customer address, and it can also have methods, such as add new customer, change customer address, and remove customer. It is a different way to view and discuss these concepts with the customer. Class diagrams look at data and process together. [Figure 8.12](#) shows a common representation of a class with the class name, its attributes, and the methods.

Classes are also related to other classes. A customer can place orders, and an order can have parts. These relationships are shown in [Figure 8.13](#). This looks very similar to the ERD shown earlier, and in reality, much of what is described in a class diagram is the same information as the ERD. However, the addition of methods is significant, as are some of the rules on how classes work together.

Behavior Diagrams—Typical models here include activity diagrams, state diagrams, and use cases. These models are perhaps the ones where the business analyst

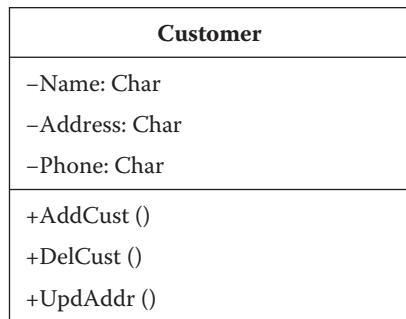


Figure 8.12 Class diagram.

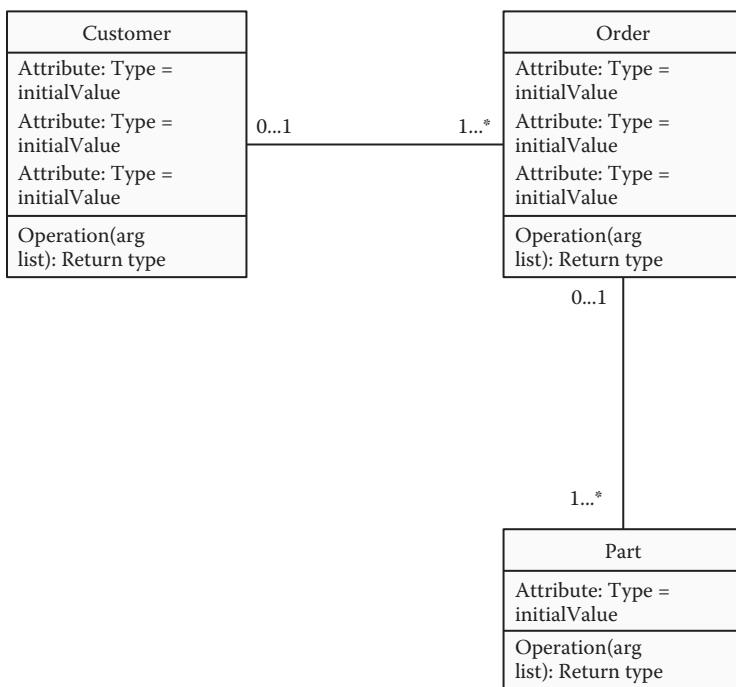


Figure 8.13 Class interfaces.

is most involved. The expected behavior of a system must be defined by the customer. Other areas, such as data and nonfunctional requirements, may be defined by standards or regulations, but behavior requires customer involvement.

Use cases may be the most popular method of documenting requirements today. Use cases were originally developed by Ivar Jacobson at Erikson. They were originally linked primarily with object-oriented development, and although they predated UML as a standard, they are now a part of this standard. Use cases have expanded outside of just the OO environment, and today many corporations use them as the standard way of documenting all types of systems requirements. Although there are business use cases as well as systems use cases, this section will deal with only the latter.

So what is a use case? A use case shows how the system will be used by an entity external to the system. That entity may be a person, an organization, or even a different system. It can also be a more abstract concept such as time. One guideline is that the system cannot start its own use case; it must always be externally initiated.

Each use case should be a stand-alone action that has a value of its own. Use cases should not be decomposed; they are relatively high-level concepts. *Place Order*, *Pay Invoice*, *Hire Employee* are typical real-world examples. This definition can cause some confusion, leading to projects detailing hundreds and sometimes even thousands of use cases. It is important to try to avoid this because it causes a lot of duplication of functionality between use cases. To start the identification of use case candidates, ask the stakeholders (including other systems) who will be using the system being developed. Then ask what those *users* will want to do. It is likely that there will be a fairly comprehensive list of potential use cases. Some of them will probably be detailed processes of the same use case. For example, customers may say that they want to place an order, change an order, and print a hard copy of an order. These are all related to *Place Order*, and can probably be handled in one use case.

Organizations often create too many use cases early on and then consolidate them as they get comfortable with the modeling approach. One technique, shared by a past student, to evaluate for potential consolidation of use cases is to ask the following three questions:

1. Are the use cases typically done by the same user? This does not mean that there cannot be two actors performing the same use case. That happens frequently; for example, both a customer and an order entry person can place an order, but it does mean that the actor who starts the use case will normally finish it.
2. Are they typically done in the same interval of time? If the use cases are *Place Order* and *Pay for Order*, determine if they happen at approximately the same time. Can you place five orders and then pay? If the actions are not in the same interval (and frequency), it may be good to split them apart (but it is not a necessity).
3. Do they both need to be implemented at the same time? In other words, if one use case is part of the project scope, must the other use case be included as well, or can it be handled manually? With the previous example of placing an order and paying for it, it may be conceivable for the customer to implement one of those functions but not the other (due to some type of constraint). If so, it may make sense to have two use cases.

If the answer to all three questions is yes, then there is a good chance that this is one use case. Although this is not an official, foolproof technique, it does help the analyst to get an understanding of what requirements belong together.

A use case has two components: the diagram and the template. Although most people may recognize the diagram, the real information and the value are in the

template. Actually, a student from one of the large U.S. auto companies mentioned that they are no longer using the diagram, just the template. The diagram gives the big picture, the scope, which can sometimes be seen in other diagrams such as context diagrams (discussed in [Chapter 3](#)). The template is where the details of the requirements are captured, at a user level, a system level, and sometimes at a business level. Both the diagram and the template will be detailed over the next few pages.

[Figure 8.14](#) shows an example of a use case diagram and some of its key components.

- Actor: Initiates the use case. It is noteworthy that all use cases must be externally initiated. The system cannot initiate any use cases. There can be secondary actors who are part of the response to the use case, but the primary actors will be the ones initiating the use case.
- Relationship: Simply shows what actor is executing which use case. Each actor can have a relationship with many use cases, and each use case may be initiated by many actors.
- Use case: The actual action that the actor is trying to accomplish. The guideline again is to have an action that accomplishes a business value for the user.
- Includes and extends: These are special-needs concepts that can be used to add flexibility to the use cases.

The expression *includes* is basically a way of calling a different use case from within the one being defined. This may be done if there is a piece of common functionality between two or more use cases. *Includes* would allow the documentation of that

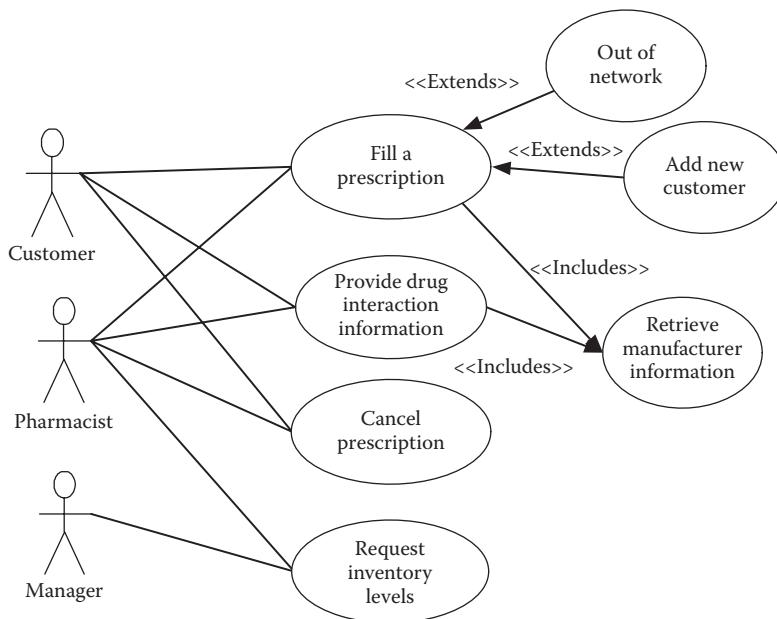


Figure 8.14 Use case diagram.

duplicate functionality once. A good example is the two use cases *Fill a prescription* and *Provide drug interaction information*. Both of these use cases will need to *Retrieve manufacturer drug information*. Instead of defining this process twice, once for each use case, an *includes* use case is identified, reducing duplication of effort.

The word *extends* allows for identification of exception criteria. These exceptions may be error conditions or just unusual circumstances. In the use case *Fill a prescription*, the normal use case may be that an existing customer is a part of this pharmacy's network. An *extends* could be added for *Out of network* processing or for adding a new customer. This would be something that is valid functionality, but not needed all the time. Some experts strongly discourage the use of *includes* and *extends*, arguing that it is really a form of decomposition. There may be valid reasons to use them on occasions, but it may also be wise to evaluate if these constructs are really needed. The *includes* seems to be the most value-added of the two, because it reduces the duplication of information, and that is always recommended.

Each use case is really a placeholder for the use case template. Although there are many examples of use case templates, such as the one shown in [Figure 8.14](#), these templates should be viewed as starting points that should be evaluated for applicability to the standard processes within each organization.

The use case template is not officially part of UML, but it is the part of the use case modeling technique on which most organizations are concentrating. Some large organizations have gone so far as no longer using the diagrams, but rather only documenting the requirements in the use case template. Technically then, this is not really part of the UML family, but it is a good example of how it is more important to customize a concept to make it work better within an organization than to strictly enforce an external standard. It should be noted, though, that if the organization is using an advanced tool such as the Rational Unified Process®, then deviating from the standard process will cause the organization to lose some of the features within the tool. An example of the use case template is shown in [Figure 8.15](#).

Some of the most common fields in the template are

- Use case ID. Each use case should have a unique identifier.
- Use case description. Overview of what the use case is supposed to do. This should be written in terms of the actor. What is the actor looking to accomplish when this use case is initiated?
- Precondition. Describes what the starting condition is. Is the user logged into the system? Have security checks been done? It helps to identify what the actor and the system must have done prior to entering this use case. Sometimes the activities described in the preconditions are outside of the scope of the project. If so, they do not need any further documentation. If they are within the scope of this project, they can be documented in other use cases or in a different type of requirements document (sometimes referred to as supplemental requirements).

General characteristics	
Use case ID	Unique identifier
Description	What is the use case intended to do
Precondition	What has happened prior to the use case being started
Postcondition	What will happen after the use case is completed
Success criteria	How will successful completion be judged
Main scenario	What will the flow be most of the time (see below, this is sometimes called happy path)
Alternative scenarios	Variations on the main flow
Exceptions	Errors
Includes	Common subroutines
Applicable business rules	What business rules apply to this use case
Frequency	How often will the use case be executed
Primary actor	Who will initiate the use case
Secondary actor	Who is involved with the completion of the use case
Security	Special security considerations for the system
Back-up and recovery	What happens if the system is not available
Assumptions and constraints	What environment is assumed What are the system constraints
Version control	What version of the use case is this
Author	Who wrote this document

Main scenario (happy path)	
Step	Action
S	Starting point of use case
1	Actor initiates use case
2	System responds
3	Actor continues the use case
4	System responds
5	Completion

Figure 8.15 Use case template.

- Postcondition. Describes what happens after the process is finished; for example, *Print receipt*, *Update balance*. Again, this may refer to something out of scope, a different use case, or a requirement documented in a different document. Together with the precondition, the postcondition helps define the boundaries of the use case. This is a critical definition because it helps focus the customer on what the starting and ending points should be for the discussion.
- Success criteria. How will the actor define a successful completion of the use case? When talking about requirements, there tends to be a lot of focus on

error conditions and alternative paths. This section is where the desired outcome of a use case is documented. In other words, what does a successful end result look like? What will the actor accomplish when everything goes right?

- Main scenario. This is the normal scenario. It describes what is expected to happen under normal circumstance. Note that it does not mean 50 percent or more of the time. Rather, it means that when the customer thinks about this process and how to reach the success criteria identified above, this is what would be described as the normal path. It also should be one of the main test cases. The example in [Figure 8.15](#) shows the interaction between the actor and the system. Because the use case must always be externally initiated, the first action must be taken by the actor. There can also be secondary actors shown, people or systems involved with the completion of the use case. However, because this is discussing a systems use case, what is shown is an interaction between actors and the system, not an interaction between two actors. The interaction can also be documented using an activity diagram or a flow chart. The scenario can have loops and decisions in it to repeat certain steps in the process or to skip other steps based on a predefined criterion.
- Alternative scenarios. Although there is only one main scenario, there could be many alternative scenarios. Alternatives deal with valid conditions that happen less frequently than the main scenario. For the use case of *Fill a prescription*, alternative scenarios may be *New customer*, *Prescription not available*, or *Customer has no insurance coverage*. These are valid conditions, the requirements for the processes must be documented, but it is not the norm. The alternative scenarios may be documented as complete scenarios, looking similar to the main scenario, or they may be partial scenarios, where the alternative actions start at a certain step in the main flow and end at a different step.
- Exceptions. These are similar in form to the alternative scenarios, but deal with error conditions. What happens when a customer orders an invalid product? What is the result if the customer does not pass the credit check? Again, as with the alternative scenarios, it is likely that there will be a substantial number of these conditions.
- Includes. This documents what other use cases are called from within this use case. Typically used when there is identical processing done between two or more use cases.
- Applicable business rules. Although business rules do not have to be enforced by the use case (even though it is desirable), it must allow for the business rules to be enforced. An example of this is the business rule, “Only the pharmacy manager can see confidential customer information.” This can be dealt with manually by not storing confidential information in the system, or it can be handled by password protecting the information. However, if the system is set up so that a clerk entering the customer information automatically will be able to see all customer information, then the business rule cannot be enforced and

there is a conflict, which must be identified and resolved. Other business rules that may impact a use case are “Prescription must be filled within 30 minutes of receiving it” and “The pharmacy must be able to process prescriptions 24 hours per day, 365 days per year.”

- Frequency. This is how often this process will be used. It is largely used to gather the nonfunctional requirements of performance. If this process is used 10,000 times per day, performance becomes critical. If it is used once every 3 months, performance is less of a concern.
- Primary actors. A primary actor is someone or something initiating a use case. It can be a person, a role that a person plays, a group of people, a system, or a time. There can be more than one primary actor. For example, the use case of *Check for interacting drugs* can be initiated by a pharmacy clerk or by a different system. Because a use case must always be initiated from the outside, there will always be at least one primary actor. Sometimes, that actor is time. A use case may be initiated by the end of the month, a tax deadline, or some other temporal concept.
- Secondary actors. Secondary actors are needed to complete the use case, but they do not initiate it. If the use case is *Check for interacting drugs*, the drug manufacturer would be a secondary actor. They are needed for the successful completion of the use case, but they do not initiate it. There can be two different types of secondary actors. The first type is needed for the use case to be completed; in this case, the drug manufacturer provides drug interaction information. The second type of secondary actor is when they are the indirect trigger of a use case. When the drug maker issues a warning for a newly discovered drug interaction, the pharmacist will enter that into the system and check for potential problems for all patients. In this scenario, the pharmacist is the primary actor interfacing with the system and the drug manufacturer is the secondary actor triggering the event.
- Security. Again, this comes back to the nonfunctional systems requirements. Is there a need for access control? Can all users perform this function? What system audit trail needs to be created?
- Back-up and recovery. Are there any special needs for recovery? Do data need to be accessible 24 hours per day? Is there a manual (or automated) process that can be used if this system is unavailable?
- Assumptions and constraints. List any assumptions that were documented as the use case was developed (such as an actor having access to the system) or constraints (must be able to connect to the ABC Company credit check network).
- Version control information. This document will go through many changes. Version control will be difficult but critical, especially to ensure that there is no scope creep as the document goes through iterations from high level to detailed requirements.
- Author. Names the person responsible for this document and can include customers and other stakeholders who provided input into the document.

A completed use case template will contain very detailed information. How does the analyst go from some very high-level objectives of a system down to the very precise details contained in the use case template? It is actually an iterative process and one of the advantages of the use case template if it is used correctly. It can be documenting the high-level requirements up front and then gradually being refined into detailed requirements. This document then, after the business analyst has completed it, is turned over to the systems analyst, who can add the solution design features to it as well. Done this way, it gives automatic traceability from the high-level requirements, all the way to the detailed design.

An example of an iterative approach when developing the use cases (for the analysis phase only), which starts with the objectives and finishes with detailed requirements, through gradual expansion of the use cases is

- High level
 - Defines scope
 - Sets the boundaries for the project
 - Only focuses on main scenario and critical alternative scenarios
 - Used to get executive commitment
- Broad
 - Gets a good description of all use cases and actors
 - Identifies scenarios, but stays at a high level
- Deep
 - Explores each scenario in detail
- Documents nonfunctional requirements

The high-level iteration should be done with the sponsor, customer management, and other decision makers. Have a clearly defined set of deliverables, including the use case diagram and a predefined subset of the use case template. The broad iteration involves the users and focuses on identifying scenarios and user requirements. The final iteration brings in the developers and focuses on the detailed systems requirements, both functional and nonfunctional. This is the session that serves as the translation and hand-off between the business analyst and the systems analyst. The business analyst is responsible for the requirements or the *what*, and the systems analyst is responsible for the specification or the *how*. Both parties must understand the documentation created here. It is technically owned by the business analyst, but it really should be a collaborative effort.

Strengths of use cases:

- They are intuitive for the customer.
- They concentrate on the purpose of the system.
- They can see both big-picture business view and detailed systems requirements.
- They give the foundation for test cases.

Weaknesses of use cases:

- There is some duplication of requirements, especially data and nonfunctional requirements.
- They may require a lot of alternative paths.
- There is the potential of getting into too much detail too fast.

Activity Diagrams—The activity diagram (sometimes called swim lane or work flow diagram) is one of the most frequently used modeling techniques. It is intuitive, the customer relates well to it, and it is excellent in defining communication points between processes and organizations. Although activity diagrams have been around for a long time, they have become more standardized and formalized under the UML notation technique. However, be careful when using a modeling tool for these diagrams. The capabilities of the modeling tool sometimes make the model more complex than is needed, and the customer may get lost in the details. Drawing tools such as Visio® have the UML notation built in, which helps to enforce the standard, but this can make the model look complex. When working on a white board or flip chart, the analyst can simplify the model, making it easier for the customer to understand. There are other standards besides UML for activity diagramming, so do not be surprised if customers or developers are used to a different way of showing the information. Activity diagrams can be shown with or without swim lanes.

[Figure 8.16](#) is an example of an activity diagram (it is actually very similar to [Figure 7.10](#) [business process flow]). This shows that modeling techniques can often be used at many different stages of the project. The swim lanes add organizational information and bring functional perspective to the diagram; but if that does not fit in with the specifics of the project, then just remove the swim lanes and show the activities as they are.

When discussing activity diagrams, the main focus is on the flow and the information going from one process to another. Most models discussed earlier in this book are not intended to show sequence, but activity diagrams are, which may make them reflect how the customers view their business.

Some of the key conventions when creating an activity diagram are

- There can be only one starting point, but there can be multiple ending points.
- People, organizations, and systems can be given their own swim lanes.
- Activity diagrams can be shown following a timeline concept or the sequence can be shown without indicating a specific time for each activity.

A closer look at [Figure 8.16](#) shows some of the strengths of this modeling technique. It shows the steps in the process in a sequential and intuitive mode. Each one of those process steps should have a document attached to it to show the detailed requirements of that process. This is best done using a standard template, similar to what was shown earlier for the use case. The diagram also shows any physical

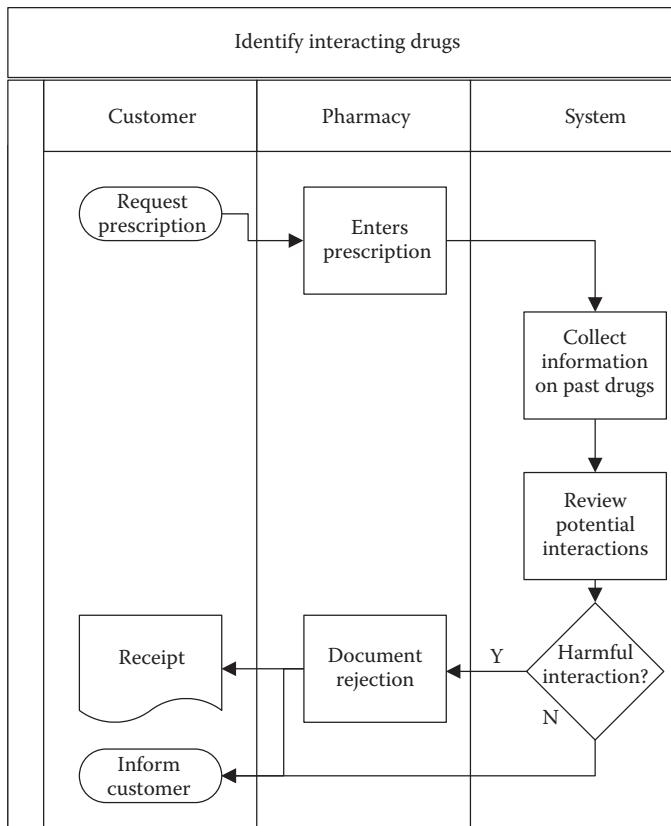


Figure 8.16 Activity diagram.

manifestation being created by the process. The documents are shown (in this example *Receipt*), and should eventually be defined in detail.

The decision points are the next notable construct. This will allow the process to branch off in different directions based on predefined conditions. On the basis of those decision points, the process may be sent back to an earlier step (looping back) or forward (skipping steps). There are also examples of a process forking (splitting into two parallel processes) and joining (coming back together again).

Of all of the models available, a well-drawn activity diagram is often the most enlightening and understandable for the customer. For the Prescription Interaction Project, the flow shows the process of checking for drug interaction. The process starts when the customer hands over the prescription and ends with a prescription verified against drug interaction and the customer informed. Note that this is a subprocess of the overall *Fill Prescription* process. It is important to clearly identify the boundaries of what is being modeled. Define the starting point as well as the end point of the process being modeled. This will help the user stay focused during the review. This model then serves as the basis for the detailed requirements gathering. Review the model step-by-step and ask questions about each step:

- What information is needed from the customer?
- Can anyone become a customer?
- How does a customer request a prescription?
- If the customer is on a prescription plan, is there additional information needed?

Of all of the models available, a well-drawn activity diagram is often the most enlightening and understandable for the customer.

This form of review will guide the user along the process and minimize excursions into nonrelated areas. Whenever users stray from the step being discussed, put their ideas on a “to be discussed later” list and gently guide them back to the activity diagram.

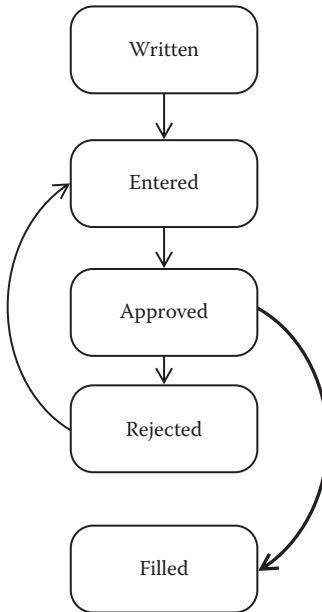
State Diagrams—State diagrams (also known as state chart diagrams) illustrate the life cycle of an entity within a business. They are part of UML, but state diagrams in different forms have been around long before UML. The intention of the state diagram is to evaluate the behavior of an entity while it is in different states of existence, such as when the entity is created, when it is going through different parts of an organization, and when it is becoming obsolete and no longer needed. State diagrams are not needed for all entities, but for entities that are being processed and changed by an organization, they can be a great tool to clarify requirements. For instance, the entity *Prescription* can have multiple states. It can be a written prescription, an entered prescription, a rejected prescription, an approved prescription, or a filled prescription. Developing the state diagram forces the customer to look at how an entity is viewed, how it can be moved from one state to another, and what some of the unique behaviors are in each state.

Review the state diagram in [Figure 8.17](#). With this example of a state diagram, it is easy to see how the model can be used as a part of the customer interview. Walk through the model step-by-step and ask the customer:

- How is a prescription written?
- Who can approve a prescription?
- How does a prescription get rejected?
- Is there a process to move an approved prescription to a rejected prescription?
- What is the difference between an approved prescription and a filled prescription?
- Should you be able to reject a prescription without approval?

Naturally, these same questions could be asked without the model, but they would probably appear more disjointed and would be more difficult to document in a structured fashion.

State diagram for prescription

**Figure 8.17** State diagram.

Interaction Diagrams—Examples of interaction diagrams include collaboration diagrams and sequence diagrams. Although some of these diagrams are used for testing and to increase overall understanding of what a system will do, they tend to fall more in the domain of the systems analyst and the developer rather than the business analyst. Classes (as reviewed earlier) interact with each other, and interaction diagrams help identify how the collaboration works and in what sequence things need to happen. However, they are usually defined at too great a level of detail for the customer. As a business analyst, the main use for these diagrams is in discussions with the systems analyst. These models are beyond the scope of this book.

8.3.1.5 Matrix Documentation Somewhere between pure text documentation and models is matrix documentation. It increases readability, can be used to validate information, and is relatively easy to implement. There are some standard matrices that are often used when documenting requirements (such as data dictionaries and use case templates); however, in addition, much of the textual requirements can be organized in some form of matrix document.

For data models, consider adding a create–read–update–delete (CRUD) matrix as shown in [Figure 8.18](#). On one side of the matrix are all of the data entities identified in the data model. On the other side are the organizational units involved with the project. In each intersection, the analysts evaluate whether this is an entity that the organization can create, read, update, or delete. This helps evaluate ownership of data, and also helps identify potential interface needs. If this project does not create an entity, but

Entity organization	Pharmacy	Inventory	Customer support
Customer	CRU		RUD
Doctor	CRUD		R
FDA	CRUD		R
Prescription drugs	R	CUD	R

Figure 8.18 CRUD matrix.

needs to use it, then the question of who does create it must be answered. Also, instead of putting organizations on one side of the matrix, the analyst can put the business processes that deal with the data. It serves a similar purpose by helping to identify what process creates an entity, what process can change it, and what process can delete it.

8.3.1.6 How to Decide Which Model to Select The systems development methodologies used will be drivers behind which modeling techniques are selected. If the development organization is more object oriented, the UML diagrams tend to be the favorite choice because they have been developed with that platform in mind. Another big factor is the knowledge and skill level of the team. If no one has experience with use cases and class diagrams, and this is a high-visibility project, then this may not be the time to try to learn these techniques.

However, the main concern of a business analyst will be to what the customer can best relate. The analyst must find a way, through models or text, to make sure that the customer agrees with the requirements that have been captured. So customer preference must also be involved in the decision process. For the Prescription Interaction Project, it is likely that the customer's experience with systems development is low. The project is an enhancement to an existing system. On the basis of these two factors, a legacy approach would make sense, *legacy* meaning that the approach would be the same as what the organization has used in the past. This would make sense both for the developers and for the customer.

The analyst must find a way, through models or text, to make sure that the customer agrees with the requirements that have been captured.

8.3.1.7 Text Documentation Very few people enjoy writing down requirements in text form. However, if after meeting with the customer for 2 hours the end result is a blank sheet of paper, then there are some real problems. It is in the text document that the analyst documents the details of the requirements. Requirements should be

- Specific. The requirement should be specific enough to allow for comprehension by all. There is always a question on how far to take this. The example “The system must allow for security” is clearly not specific enough. What type of security? Security of what? By changing the statement to “The system must

allow for multiple levels of access to restricted data,” it is greatly improved. It becomes a judgment call whether to further define it or to leave it for the design team.

- Unambiguous. Does it mean the same thing to different people? “The system must be fast” could mean different things for the developer and the customer. The statement needs more detail. “The system must have an average response time of 3 seconds during peak hours, from 9:00 a.m. until 3:00 p.m. Greenwich Mean Time” would give a much more measurable requirement.
- Verifiable. This may be the most important one. All requirements must be verifiable after implementation; if not, they are meaningless. It may be too early to determine exactly how it will be verified, but when looking at the requirement, it should be clear that we can verify that the requirement has been met. The requirement “The system must make the customer happy” is probably not measurable, but the requirement “The system must allow the customer to enter a prescription in less than 1 minute” is.
- Traceable. All requirements must be traceable from a business need to user requirements, to system requirements, to specifications, and finally to the finished product.
- Consistent. All requirements must be consistent with the other requirements. If requirement A asks for a wireless interface and requirement B states transmission must be over landlines, then there is an inconsistency that must be resolved before the business requirements document (BRD) is turned over to the development team.
- Agreed to. Both the customer and the developer must agree to the requirements. It is the customer’s prerogative to find a different developer if they see fit, but no developer should take on a requirement that he or she does not think can be implemented.
- Realistic. Even if the customer wants response times faster than the speed of light, that does not make it realistic.

It is also important to use good writing practices when writing the requirements. Some examples of good writing habits are

- Use proper grammar and spelling. Sometimes when reading documentation, it is surprising to see that the author has not spell-checked the document. However, spell-check sometimes causes the author to assume that because the tool does not flag anything, there are no errors. It is very difficult to catch one’s own mistakes. Always have a different person proofread the document before it goes to the customer. A poorly written document will leave a bad impression with the customer and may cause the customer to question the competency and professionalism of the analyst.
- Write in the active voice. Do not write “The user would like to be entering order information” but rather “The user enters order information.”

- Define terms and abbreviations in a glossary. There may be terms that seem obvious to the person writing the document, but for others, the terms may be totally unknown or have a different meaning. Acronyms such as DBA may mean *database administrator* in the information technology (IT) organization and *doing business as* for the customer. For the Prescription Interaction Project, something like FDA may cause confusion. Although clearly a related project entity in the United States (Food and Drug Administration), in a different part of the world it could be a different regulatory agency.
- Remove ambiguity. Sometimes a word can mean many things and must be clarified to be understood by all stakeholders. A simple word like *system* can cause confusion: is it the computer system or the business system that is being talked about?
- Use the customer's language. Because the project is likely to eventually lead to a systems development effort, it is easy to start early in the process with technical terms. Avoid this, and talk the customer's language. It is unavoidable that the customer will need to learn some systems terms, but for the most part it should be the developer learning the customer's language. This is especially important when using development tools. These tools often enforce standards that make sense from a developer's view, but which can be very confusing to the customer. For example, the customer may state a need to track the manufacturer. The standard in the data modeling tool could be *Presc_Mfr*. This is obviously not intuitive for the customer and should be avoided, except for customers with prior exposure to these formats.
- Organize in single, focused, and short sentences. Rambling, never-ending sentences are difficult to understand. They tend to contain so much information that it is difficult to see what the real purpose or focus is. Decompose the requirements so that each requirement has a clear focus and a real purpose. Try to ensure that the successful completion of a requirement can be verified with just a few test cases. There will need to be more test cases for errors and alternative paths; however, if there are a large number of test cases needed just to verify successful completion, review the requirements to see if they should be decomposed. For example, the requirement "The system must verify valid zip code, compare against past prescriptions, prompt the customer for other medications not in this system, possibly from other pharmacies, and pull the latest drug interaction information from the manufacturer and any controlling government agency" is too long, too complex, and lacks focus. Split it up into short sentences where each requirement has a clear focus.
- Avoid creativity. Writing a BRD is not the same as writing the Great American Novel. It should communicate clearly. Strive for consistency and avoid synonyms. If it is called *drug* in one place, use that term consistently. Although *medication* and *product* may mean the same thing in the pharmacy, when alternating between the terms in a requirements document, it will cause confusion to the developers when trying to figure out the difference between a *drug* and a *medication*.

8.3.2 Verify the Requirements

Most testing done is for product verification, which means making sure that the system does what the requirements said it should do. It is primarily a tool to see if the developers correctly understood and implemented the BRD. Traditionally, if the product passed the test, but the customer is not satisfied, the developer (and business analyst) can go back to the customer and say, “You got what you asked for.” That may be a valid argument, but it is no longer good enough when developing a product. If an organization is to be successful in the long run, it must not only give the customers what they ask for but also give the customers a product that will solve the problem they had in the first place.

There is a general shift in the industry today away from product verification as the main goal, even though it is still important, and toward product validation, which is much more difficult, but also more critical and much more rewarding. To validate a product, the original objectives must be evaluated and a determination made if the product met those objectives. Were the problems and constraints dealt with effectively? Were the cost savings achieved? Did the market share increase? Did the organization become Sarbanes–Oxley* compliant? Organizations that successfully verify and validate their products will have a strong competitive advantage and are likely to win more business from their customers.

Organizations that successfully verify and validate their products will have a strong competitive advantage and are likely to win more business from their customers.

Look at the following example based on the Prescription Interaction Project. The customer has a need to minimize lawsuits and unnecessary deaths or illness resulting from prescription drug interactions. To do this, the customer stated the requirement, “All new prescriptions must be checked against other prescriptions on record for the customer in order to identify interaction problems.” Although this sounds good, the business analyst must dig deeper. How severe is the problem today? What causes the problem? How much of the problem will go away if this requirement is implemented? The business analyst must go to work quantifying the problem. Some causes may be due to either a customer using a spouse’s prescription or the customer having old medication from a different pharmacy. Neither of those two scenarios would be resolved with the requirement as stated. So, in this case, while verification may be met, validation will not be.

What would be a better requirement? There may need to be an industry-wide database that will show prescriptions across all pharmacies. Maybe the requirement needs to be expanded to include checks against family members. Maybe there needs to be a checklist added for questions that the pharmacist asks the customer. The point here is that what looks like a good idea initially may not actually solve the customer’s

* Sarbanes–Oxley is legislation that resulted from the fall of Enron and is an example of legislation that becomes a major driver of many IT projects.

problem. If the customer has just spent a large amount of money without having the problem fixed, that is ultimately going to have an impact on customer satisfaction.

8.3.3 Validate the Requirements

After weeks of eliciting requirements and days (or more) of documenting them, the final step in the analysis phase must be to validate them. There is a more detailed discussion on validation in [Chapter 9](#), including what to include in the review package for the different stakeholders. Both models and text must be validated, and that will normally be done in some type of walk-through. It is especially important for the users to be educated on the models that will be used in the walk-through. Although some of them may have been seen and used during the elicitation of the requirements, be careful not to assume that each reviewer has previously seen the models, has been trained on them, or even remembers how to read them.

8.3.4 Define Requirements Architecture

When we organize requirements, we normally use some form of taxonomy. *Merriam-Webster's Collegiate Dictionary* (eleventh edition) defines *taxonomy* as “orderly classification” The first famous taxonomy, developed in 1735 by the Swede, Carl Linne, classified plants and animals. A good taxonomy focuses on finding common elements to create a structure that makes the comprehension of a knowledge area easier. A taxonomy can be related to different attributes, such as color or speed, or to different sources, such as governments or users.

The Software Engineering Institute® (SEI®) has developed a risk taxonomy (TR06.93) for software development, which is available from the SEI website. It basically breaks the system down into classes, elements, and attributes for the purpose of evaluating risk, as seen in [Figure 8.19](#).

There are three classes identified:

- Product engineering
- Development environment
- Program constraints

Each class is then further decomposed into a number of elements. One of the elements within product engineering is requirements, and this element is further decomposed into a number of attributes.

- Stability: Are the requirements changing throughout the process or are they well known, understood, and static in nature?
- Completeness: Are there still areas that have not been explored? Are external interfaces well understood? Does the customer have expectations that are not clearly recognizable in the requirements document?

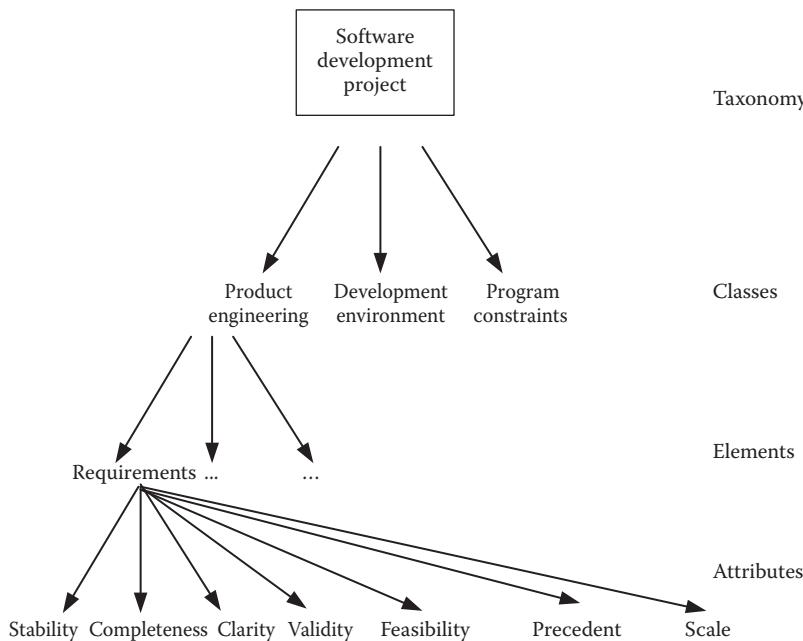


Figure 8.19 SEI risk taxonomy.

- Clarity: Are you able to understand the requirements as written? Will different stakeholders interpret them the same way?
- Validity: How are the requirements validated? Are there things in the product that the customer does not want?
- Feasibility: Are these requirements possible to implement? Have any feasibility studies been done?
- Precedent: Are these requirements that have been previously implemented in this organization? Have they been implemented anywhere else?
- Scale: Is this project similar in size to past projects? Does the organization have the capability to implement a project of this size?

In this example, the hierarchy is intended to evaluate the quality of the requirements gathered and the risk associated with them. The business analyst should use this type of checklist during the requirements checking, when the BRD is finalized and reviewed.

In its basic form, a requirements taxonomy is little more than a checklist that helps the analyst to ask the right questions and ensure that all key areas of the business have been covered. Using the requirements element of the risk taxonomy above, the checklist would simply say

- Are the requirements stable?
- Are the requirements complete?

and so on.

In addition to covering different areas of the business, or the technology, the checklist also encourages the business analyst to delve more deeply into each part of the taxonomy. For example, a requirements category of *security* may list a requirement such as “The system must limit access to sensitive financial data.” This should prompt the analyst to ask “What is the definition of sensitive financial data?” and “What other types of security requirements are needed?”

Building a classification system for an organization takes a formal process and a repository for lessons learned. **Figure 8.20** shows a process for building and improving such a system. Step 1 is to identify any existing ways of categorizing requirements based on the type of project being developed. Often there is no official standard for categorization within an organization; however, that does not mean that there is no categorization being used. If no formal document exists, such as a BRD template, then check with the people who have experience with projects in the organization. Ask what types of requirements are important. Are there internal, corporate requirements? Are there any external requirements? What organizations will typically provide requirements? By asking these types of questions, a rudimentary structure for a taxonomy (or classification system) will start to appear.

In step 2, the actual requirements collected are put into their respective categories. Put any requirements not fitting into the categorization system into an *Other* bucket. At the end of the project, these can be used to identify potential new categories for the future. Some requirements may fit into multiple categories. Decide on one and then make a note in the other category or categories referencing the selected category. Why? Often, different categories will be assigned to different groups or individuals for analysis. Adding the reference without putting the requirement itself in multiple categories will help minimize duplication of effort, while still keeping the lines of communication open.

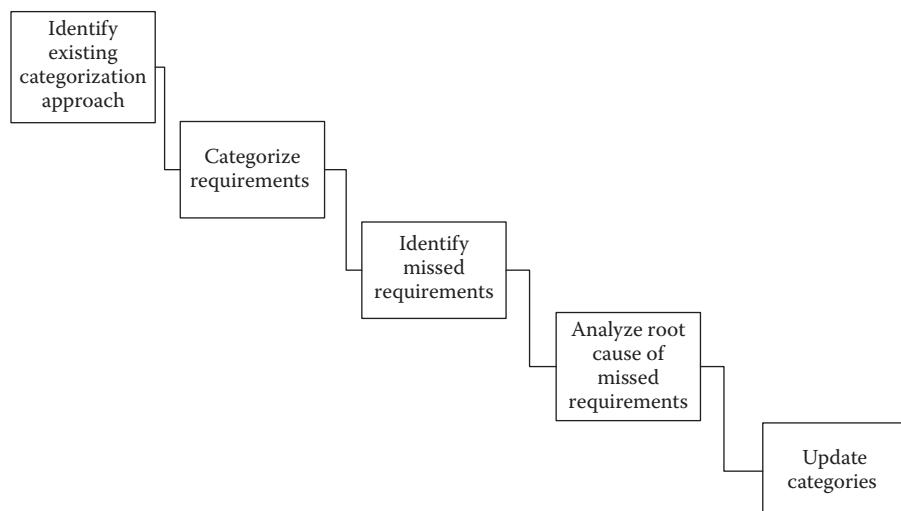


Figure 8.20 Building of a classification system.

Step 3 is done during reviews of the BRD, during testing, and ideally in post-project reviews. It identifies missed requirements, which will be used for many types of process improvements; here, they will be used to evaluate root causes related to requirements categorization.

In step 4, the root-cause analysis is then performed to try to identify why a requirement was missed. Was it due to poor elicitation techniques? Or, was it due to not asking about certain types of requirements? If so, these requirements will be taken to the next step.

Step 5 then updates the categories of requirements based on the input of the root-cause analysis in the previous step as well as the content of the *Other* bucket mentioned in step 2. This newly created list will act as a repository for lessons learned for future projects.

By the time this process has been used on multiple projects, the classification system will have become more helpful and actually assist in standardizing the organization's approach to requirements gathering.

8.3.4.1 How Much Detail Do You Need? When looking at requirements taxonomies, it is easy to get the impression that the more detail there is the better. That is not true. There should be as much detail as is needed, not a bit more. Naturally, the difficulty is to judge where that line is. With the standards and processes from the PMI®, SEI, and IIBA in their back pockets, many organizations are now developing large, detailed, sophisticated, and utterly confusing requirements documents.

There should be as much detail as is needed, not a bit more.

Karl Wiegers writes in *More about Software Requirements*, “The requirements may be vague, but the product will be specific” (p. 17). He uses the statement to emphasize the need for precision in requirements definition, which is of course needed, but the statement also hints at another aspect of requirements gathering. Who should determine the specifics of the product? Does it have to be the customer? What if the customer does not care? Or, what if the customer is clueless?

Development organizations often have the attitude that if customers working with developers do not know what they want, or do not understand the business, then the developing organization must push to get different customers involved to ensure that all the requirements come from the customers. It is true that there must be safeguards against making assumptions about the customer’s competency, but it is also true that many times customers do not know what they want, do not want to be involved with the development process, and should not have to be involved. The customer’s expertise is not systems development or process improvements. The business analyst, together with the developers and the customers, must make an assessment up front on what requirements come from the customer and what requirements come from other sources. Compare it to buying a vehicle. Going to a car dealer, most people would not

expect to have to tell the dealer that they want four wheels, a steering wheel, carpet in the trunk, seat belts, etc. Although it is true that the customer would notice (and complain) if those items were not there at the time of delivery, the customer has a right to assume that the product developers will do some thinking on their own. Some organizations struggle with taking on that responsibility. They create hundreds of pages of requirements, overwhelming the customers and forcing them to think about things that they should not have to think about.

Many times customers do not know what they want, do not want to be involved with the development process, and should not have to be involved.

A good taxonomy can help with deciding what the customer should or should not be involved with from a requirements-gathering standpoint. If the customers just want a good sales-reporting system, they should be able to tell the analyst what type of sales reports they want at a fairly high level (business requirements level), and then expect the analyst to work out the details. The analyst can research what packages are available, what other organizations do, and come back with some ideas that the customers can verify as being on the right track.

For the Prescription Interaction Project, the customer should be able to state a need for a system where staff can enter all the prescription information, check inventory, and look up customer information, but the customer should not necessarily have to provide every detail of how the system should interface with other areas, or what the look and feel of the system should be. That can be handled through organization standards and research by the business analyst and the developing organization. On the other hand, if the customer has strong feelings about what the system should look like, then he or she needs to be involved in the detailed definitions of the user interface.

So when trying to define how detailed the requirements-gathering effort should be, keep the following rule in mind: the users have a responsibility to provide the requirements for the things they care about, but the developer has the responsibility to be able to create a good solution for the areas that the users do not care about.

In general, different approaches will require different levels of details. If the effort will result in buying a package, something available off the shelf, then the requirements should be defined at a higher level, with a focus on what the system should do, rather than what it should look like or how it should flow. Changing the look and flow is normally not a good idea when selecting a package. If the user is not willing to live with the basic look and feel and flow of the package, it would be better to go to custom development.

Custom development can be broken into two categories for the purpose of this discussion. If the system will be developed in-house with the analyst, developers, and the users co-located, then there is less need for a lot of detailed requirements. Unclear areas can be worked out as the effort progresses. However, if the development will be

outsourced, the customer is dispersed globally, or the development team is in a different location, then the requirements document must be much more formal. This is especially true if the development team is offshore, involving time zone, language, and cultural differences. The key here is communication. How easy (and how likely) is communication between the parties? If ongoing communication is rare, then the development effort must be more rigorous.

8.3.4.2 Stakeholder-Based Classification One of the best ways to classify requirements is by stakeholders. By documenting whose requirements have been captured, it is easier to see if any stakeholders were overlooked. This helps with the prioritization of requirements, by identifying which stakeholder is the most important (as defined by the sponsor). It also helps with assigning responsibilities for sign-off.

There are two general categories of stakeholders:

- External: government, customers, vendors
- Internal: management, users, other departments

Within each high-level category, add the people and organizations whose requirements may be pertinent for the project. As an example, subcategories of external stakeholders may include

- Local government for building codes
- The Internal Revenue Service for tax laws
- The FDA for drug laws
- Major customers who have requested services
- Vendors who will provide part of the product

Internal stakeholders may include

- Project sponsor
- Product manager
- Purchasing department
- Order entry clerks
- Pharmacist

Once the initial list of stakeholders has been captured, that list should be reviewed with the sponsor, project manager, and other key decision makers. There are three main goals to be accomplished by this review. First, look for completeness. Are there any other key stakeholders who should be added to the list? Second, evaluate the priority of the stakeholders. Not all stakeholders are of the same importance and when prioritization of requirements is needed, it helps to know whose requirements matter the most. The third reason for the review is to identify representatives from each stakeholder group. The earlier they are identified and notified about their participation in the requirements effort, the more likely it is that they will be available for the requirements-gathering sessions.

Because these representatives will actually be the people from whom the requirements will be gathered, the process of buy-in and commitment must start as early as possible.

8.3.4.3 Sequence-Oriented Classification Sequence-based classification recognizes that different levels of the organization have different requirements of the product being developed. It also recognizes that these requirements should be captured in a certain sequence, from high level to detailed, and from the upper levels of the organizations to the end users. For top management, these may be captured as goals and objectives; for middle management, it may be operational improvements; and for the users, it may be what they need to get their job done. The following is a description of a possible sequenced-based structure, followed by an example of the corresponding organizations or people who will provide the different categories of requirements for the Prescription Interaction Project:

- Regulations, industry standards, and corporate policies. These may be mandatory or they may be recommendations. They are often documented as constraints on the solution. The IIBA does not currently list this as a category. The assumption probably is that it is included in the assumptions and constraints category mentioned below. Examples are the FDA for drug information, the Federal Communications Commission for how data can be transferred, and the local government for reporting requirements.
- Business requirements. This category identifies what management is expecting out of a project or product. Business requirements are gathered at different levels. It is a good idea to subcategorize into
 - Strategic—Where is the business heading? This may have been documented in the product vision or within business goals and objectives. Understanding the strategic requirements will tell a lot about executive expectations and can also drive the type of questions the analyst should ask. If the strategy is global expansion, it is likely to create a different set of interviews than if the goal is cost reduction.
 - Tactical—How will the strategic goals be reached? Often the project is part of a corporate initiative that should be tied back to the strategic vision of the organization. The analyst needs to understand what those initiatives are and how this project fits in with them. This is often focused within segments of the business in support of the strategic goals. So at this level, the customer could be the director of marketing, the operations manager, or the person in charge of data entry. The focus for the business analyst is to find out what these people need to be able to support and operate within the framework of the strategic requirements. The requirements being given here must be traceable back to the strategic level.
 - Operational—How should the business operate on a daily basis? This will include requirements dealing with productivity, training, process

performance, security, and other requirements important to the first-line manager in the organization who will be using the system. It will also include information about what reporting and outputs that manager is looking for. This is targeting the first-level manager, the supervisory level. It could be a supervisor over the group whose people will be using the system, or someone who needs to be informed about impacts on the business by the system.

- Stakeholder requirements. These are requirements needed by the person who will actually be using the system being developed, or who needs to perform some of the tasks manually if that is what the eventual solution calls for. For most users, the system is a tool to help them perform their jobs. Different users are looking for different things. Some need higher productivity, some need more security, and others just need a repository of information. When capturing user requirements, do not be too narrow in the elicitation. The focus should be on what the users need to do in their jobs, not what they need the system to do (that will be captured in the next point). This means that some of the user requirements that are captured will never be implemented within the system. Depending on the solution selected during the design phase, some of these requirements may be automated, whereas others may be handled manually. For example, the requirement of “User must be able take an order” will likely be automated, and “User must verify customer’s home phone number” will likely be a manual job requirement, dealt with outside the boundaries of the system.
- Solution requirements. What will the system do to assist users in their jobs?

This can be subcategorized into

- Functional requirements—These identify the core reason for the system. These requirements concentrate on what the system does and tend to be what first comes to mind for the customer. Examples are “System must allow for entry of orders” and “System must print out a receipt.” Functional requirements are things that the business would need to do even if the system was not there and the process was manual.
- Quality-of-service requirements—This is an IIBA term. In other standards, this is often referred to as nonfunctional requirements or supplementary requirements. These requirements are really the characteristics of the system. They are sometimes difficult for the customer to determine because the requirements relate to the system more so than the business problem that the customer is trying to solve. In general, if there were no system, most of the quality-of-service requirements would not apply. Quality-of-service requirements come in all different flavors and can be subdivided in many ways. Some common subcategories include
 - Environmental—Where will the system be located? Are there weather-related concerns? Are there concerns about infrastructure? Are there legal issues? This can deal with accessibility outdoors, or legal restrictions on where a certain piece of hardware can be placed.

- Interface—Most systems today interface with other systems. What are those systems and what type of interface is needed? The interfaces may be internal or external, new or existing, depending on how the current effort will change business processes.
- Operational—This can include the number of operators, operational environment, and access issues. Does the system need to be accessible without using a keyboard? Does it need to have voice recognition?
- Performance—at this level, the requirements deal with system performance. Process performance is included under business requirements, but here the focus is on expected systems responses. How much time is acceptable for the system to look for potential drug interactions? This will potentially drive the design later on. If the requirement is for response times of less than 1 second, then most of the information may need to be stored locally.
- Privacy—Is there sensitive information that must be kept private? Who should have access to what information? How should unauthorized use of the information be prevented? Are there legal issues with using the data in a way that may not be visualized by the customer?
- Safety requirements—in what environment will the system be used, and are there any safety concerns related to that? As an example, if a navigational system is developed, how do you ensure that it is not impeding the safety of a driver using it?
- Security—who should have access to the system? Who can see data? Who can update data? This may overlap with privacy requirements, which brings up a point worth remembering. Quality-of-service categories are often overlapping. It is not terribly important to put a requirement in the right category; the focus should be on looking at a project from different angles to get a complete picture. Whether the requirement has been classified as a safety versus a security requirement is less interesting than the fact that it has been discovered.
- Training—What training is needed for users, support personnel, and other stakeholders? This includes formal training but also *train-the-trainer* approaches and self-directed learning.
- Assumptions and constraints—These are predetermined characteristics of the solution. Assumptions are educated guesses. The right answer is not known; however, in order to make progress, an assumption is made and work is done based on that. For requirements, an assumption may be that “the current infrastructure will be able to handle the new workload.” As the requirements gathering progresses, this may prove to be false and, if so, this will lead to a change control item. All assumptions carry a risk, but it is impossible to make progress capturing requirements unless some are made. It is important to document them and get agreement from the

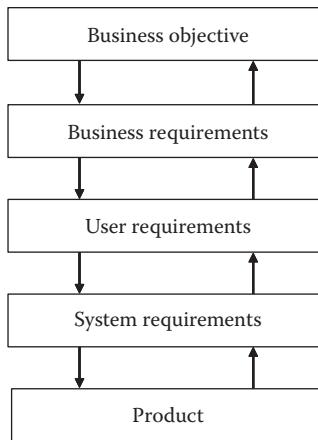


Figure 8.21 Traceability.

key stakeholders. Constraints are limitations. A constraint may be that “the new system must use the existing infrastructure.” This is a requirement, and it must be captured and communicated to the developers.

- Transition requirements. These deal with the capabilities that the solution must have for the transition into the solution. How much downtime can the business afford? Is there a need to run two systems in parallel? Does data need to be converted? Examples are “All historical data must be converted and available for the new system when it is installed,” and “A system back-out plan must be developed that allows the business to revert to the old system with a maximum downtime of 15 minutes.”

All of these requirements should be traceable to and from each other, as seen in Figure 8.21. This is a key concept of scope control. If a requirement is found at the system level and cannot be traced to the levels above, there should be an evaluation of why the requirement exists. Is it scope creep? Did a higher-level requirement get missed? Quality-of-service requirements are often traceable to regulations, industry standards, and corporate policy requirements, and functional requirements tend to trace back to user and business requirements.

The following is a review of an example of this traceability for the Prescription Interaction Project. For each requirement level, it identifies who the requirements provider is and gives an example of a requirement that belongs at that level.

REGULATIONS, INDUSTRY STANDARDS, AND CORPORATE POLICIES

Source: FDA

Requirement: All prescriptions filled must be checked against the FDA recall database for any open recalls. Any prescription with a recall issued against it

should result in a rejection of the prescription, a notification to the doctor, and a notification of the attempt to the FDA.

STRATEGIC

Source: Executive steering committee

Requirement: The Prescription Interaction Project must reduce lawsuits based on known drug interactions by 75 percent. Currently, C.V. Green is the target of 100 lawsuits each year and is paying \$10,000,000 in compensatory damages. Both of these measurements must be reduced by 75 percent.

TACTICAL

Source: Sales and marketing

Requirement: The Prescription Interaction Project must integrate information from all C.V. Green locations to discover potential prescription drug interactions with medications purchased at a different C.V. Green location.

OPERATIONAL

Source: Pharmacy manager

Requirement: All attempted overrides of drug interaction warnings must be approved by a senior pharmacist. The senior pharmacist can, after consultation with the doctor or the FDA, override a potentially harmful interaction when a decision has been made that the risk of taking the drug is less than the risk of not taking the drug.

USER

Source: Pharmacist

Requirement: Must be able to enter a customer's prescription drugs purchased from other pharmacies in order to search for drug interactions.

SYSTEM—FUNCTIONAL

Source: Pharmacist

Requirement: The system must be able to print out a complete set of drug interactions found for a specific prescription. This includes drugs available locally and all applicable government agencies.

SYSTEM—QUALITY-OF-SERVICE REQUIREMENT

Source: User and developer

Requirement: The pharmacist must be able to enter a prescription for one drug in less than 60 seconds. This assumes that there are no drug interaction warnings and that the customer is an existing customer, already in the system.

SYSTEM—ASSUMPTIONS AND CONSTRAINTS

Source: Organizational standards

Requirement: All data must be stored in the corporate Oracle™ database, accessible from all locations.

SYSTEM—IMPLEMENTATION REQUIREMENT

Source: Customer

Requirement: Past prescriptions must be converted and available on the day of implementation. Prescriptions from the 24 hours before implementation can be converted and added to the system within 4 hours of implementation.

Similar to the stakeholder classification system discussed earlier, the sequence-oriented classification is likely to need different people involved with giving the requirements for each level. Start with regulations, industry standards, and corporate policies. Then move down through strategic, tactical, and operational requirements, and continue with the user, and finally the system. It is likely to be an iterative process, though. While progressing down the list, it is likely that it will be necessary to back up and go through the process multiple times.

In this structure, the requirements should be traceable from top to bottom (and vice versa). All system requirements must come from somewhere. It could be based on a user requirement, or a business requirement, or maybe even an external requirement, but it cannot just appear at the system level. If it does, then there is scope creep or a missed requirement at a higher level. There are two ways of solving this issue. The customer can decide that either this is a valid business requirement, which means it should be added to that category, or it is not a valid requirement, which means that it should be eliminated as a system requirement.

8.3.4.4 Purpose-Based Classification In this type of classification, the project is classified into requirements based on the type of functionality desired. Is it behavior related? Does it relate to security? Does it define the features needed? Is special reporting needed?

Relating purpose-based categories back to what was reviewed earlier for functional requirements, one way to identify the purpose-based categories is to ask different stakeholders what they want to accomplish with the system. Another way is to think of it as functions that the business needs to perform, regardless of whether they are to be automated. Examples include

- Enter order
- Pay bill
- Collect customer information

From a nonfunctional requirements view, purpose-based categories deal with the things that are needed because of the system. Categories such as training requirements, security requirements, and performance requirements fit in here. They are really the same as the quality-of-service requirement category mentioned above.

Purposed-based requirements are often tied in with specific job functions. What does the order clerk need? What does the purchasing department want? Does finance have any requirements related to this project? The advantage of this approach is that it is easy to set up an interview with the representative for that job function and document the result in its own category. However, there are often conflicts between different job functions about what they would like to have the system do, and those conflicts can be difficult to find in this type of approach.

Another purpose-based approach is to focus on key modules of the product. What are the reporting requirements? What are the data entry requirements? What are the query requirements? This is a common and effective approach when the system being developed is large and complex and contains natural modules.

8.3.4.5 Combining Structures The business analyst must always keep in mind the purpose of using a taxonomy. It is used as a communication tool to help identify a complete set of requirements and to group related requirements. It organizes the thoughts of the analyst and of the customer. Identify as many structures as possible and review them to find applicability to the current business. There is probably more than one classification approach that will work well within the business area, but do not assume that any classification system will work. The classification system selected should reflect the customer's view of their business in a manner that is easy for them to follow. Customers can be frustrated if they are forced to fit a requirement into a category with which they do not agree. Do not argue with the customer; just add a category and customize the classification system for the project. Then, as a part of lessons learned for the project, evaluate if it is a value-added category that should be kept, or if it is just a one-time event. Look at the taxonomy as any other tool that must be customized for each project and realize that the main drive behind having a strong taxonomy is to improve communication.

There is probably more than one classification approach that will work well within the business area, but do not assume that any classification system will work.

8.3.5 Define Design Options

The customer is often accused of not being able to think outside the box and of being too rooted in today's environment to be able to accept creative and innovative solutions. Although that can be true, the same can often be said about the developers and the business analyst. The easy path tends to be that when a potential solution is found, it is presented to the customer, and then the team marches to that beat. Then, at the end of the project, the customer comes back with some other solutions such as better ways of doing the job, new processes, new tools, and new thinking. By spending a

little bit more time up front, brainstorming, doing benchmarking, and just exploring different options, a lot of grief can be avoided later on. Note that this does not mean that the new ideas are always better; however, if the new ideas have been reviewed and considered, there will be a higher degree of confidence that the business analyst has been looking out for the customer's best interest.

Some options to consider include the following:

- Change the business practice. IT people tend to view IT as being the solution to all problems. This is natural. Most professions feel that they have the answers to any problem identified. If you are a surgeon, you will cut; if you are a chiropractor, you will crack; and if you are an IT developer, you will automate. Be wary of this. Many problems will not be solved by implementing a system; the system will only provide a faster way to reach the problem.
- Enhance the existing system. If the current environment is basically working, do not redo everything. Often a simple enhancement solves a problem faster and cheaper.
- Write a new solution. This tends to be the default approach, at least for a large problem. It does mean that there are fewer constraints based on the current environment, but it also means higher risk.
- Outsource writing a new solution. Similar to the previous item, but the project outsources development to mitigate risk.
- Buy an existing solution. Do not reinvent the wheel unless there is a reason. If a package meets the requirements, then this is often the best approach.
- Do nothing. This should always be considered. What will happen if nothing is done? How bad is the problem? How bad will the solution be?

Some of these options can also be subcategorized. For example, to write a new solution or buy an existing solution may have subcategories reflecting different technological alternatives. Does the customer want to focus on existing technology? Or should it be more of a leading edge initiative, where new technology will be implemented?

Once a number of potential solutions have been identified, these solutions should be evaluated based on a selected set of criteria. Work together with the customer, the sponsor, and the organization to determine what the driving factors should be. Also look at what weight should be assigned to each one of those factors. Typical factors may include

- Timeliness—If there is a strong urgency to solving the problem, enhancing an existing system or buying off-the-shelf software may be more realistic.
- Cost—if cost is a major driver behind the decision, changing the business practice or doing nothing may be options.
- Organizational competencies—if the organization is short on technical expertise or there are competing business priorities, then look at outsourcing or changing the existing system.

- Available products—Are there products available for this area? If not, the option of buying off the shelf goes away.
- Competitive business practice—if this problem deals with the core of the customer's business and there is a need for a competitive advantage, it is more likely that the organization will need to develop the product.

Once a comprehensive list of evaluation criteria has been identified, they should be assigned weights. This can be done as a subjective process by just asking the customer to give each criterion a weight from 1 to 10. Or, if there is a need for deeper analysis, use a technique such as forced pair ranking. The example of this, first introduced in [Chapter 5](#), is shown again here in [Figure 8.22](#).

On the basis of this, the total weights for the four requirements are

1. FDA interface:	6
2. Manufacturer interface:	12
3. Wireless:	8
4. Web access:	4

The weights created out of the forced-pair matrix can then be used to evaluate potential alternatives. The business analyst may be asked to evaluate some package solutions against the requirements to see which one would fit the best. This time a matrix is created where the requirements (or selection criteria) go on top of the columns and each row represents a potential solution, as seen in [Figure 8.23](#).

Each requirement carries the ranking weight from the forced-pair ranking in [Figure 8.22](#). Each alternative solution is then evaluated to see how well it meets the requirement. In this case, the evaluation consists of

- 0 = Solution does not meet requirement.
- 1 = Solution does meet requirement.

1. FDA interface	1. FDA interface			
2. Mfg interface	2	2. Mfg interface		
3. Wireless	3	1	5	3. Wireless
4. Web access	4	0	4	4. Web access

Figure 8.22 Forced pair filled in.

<i>Solution/ requirement</i>	<i>FDA interface weight = 6</i>	<i>Manufacturer interface weight = 12</i>	<i>Wireless weight = 8</i>	<i>Web access weight = 4</i>	<i>Total</i>
Buy product ABC	0	0	1	1	12
Buy product DEF	1	1	0	1	22
Build product	1	1	1	1	30

Solution score: 0 = does not meet, 1 = meets

Figure 8.23 Alternative solutions.

That evaluation can be made more sophisticated with objective criteria for what it takes to get a certain rating. For example, if one of the requirements is, “Must cost less than \$100,000,” a solution costing \$100,000 or less may get a 3, a solution costing \$100,000 to \$125,000 gets a 2, a solution costing \$125,000 to \$150,000 gets a 1, and anything above \$150,000 gets a 0. This allows for more granularity in the selection process. By predetermining the scales used, the process tends to become more transparent and objective.

The process of identifying and evaluating the alternative solutions should result in a formal document, presented to the decision makers, documenting each alternative, with an evaluation of how well it meets the selection criteria as well as with an overall analysis of impact on the business and the project. Some of the decisions made here may have a long-lasting impact on the organization. As an example, deciding to go with a package solution for a significant business area is likely to drive other development efforts later on, from both positive and negative standpoints. From a positive view, it may make it easier to implement other packages from the same vendor down the road; from a negative standpoint, it may limit development flexibility later on, tying the organization into the functionality and integration limitations of the package.

Each one of the solution criteria also has a different risk ranking associated with it. [Figure 8.24](#) shows where on the risk spectrum the different criteria discussed would fall.

The risk analysis should be done from two perspectives: (1) organizational risk, which is the domain of the sponsor and the executives, and (2) project risk, which is the domain of the project manager.

Many alternative solutions will be based on different technologies and as such would not fall directly in the domain of the business analyst. It is often emphasized that the business analyst should focus on *what* the product should do and leave the *how* it will do it to the systems analyst or the infrastructure group. However, it is still the responsibility of the business analyst to evaluate the impacts and risks of each

<i>Requirement</i>	<i>Organizational risk</i>	<i>Project risk</i>
FDA interface	Low: well defined	High: external dependency
Manufacturer interface	Low: well defined	Medium: external dependency
Wireless	Medium: infrastructure not in place	Low
Web access	Medium: limits access control	Medium: Low internal skill set

Figure 8.24 Requirements and risk.

solution option, and clearly communicate those to the customer. Some of the factors that the business analyst may need to bring back to the customer are

- Gap analysis between the requirements and the capabilities of the technology selected
- Short- and long-term cost impact from technology selection
- Potential organizational value of different technologies (both within the project boundaries and throughout the organization)
- Organizational constraints

It is still the responsibility of the business analyst to evaluate the impacts and risks of each solution option and clearly communicate those to the customer.

Although business analysts are not the drivers of selecting a technology, they do need to be able to inform the customer about it. So again, it can be seen that the job of the business analyst is very broad. The analyst will need to have an understanding of most aspects of the project, even the areas with which he or she is not directly involved.

8.3.6 Analyze Potential Value and Recommend Solution

Once the analysis of the alternatives has been completed, a decision package should be assembled for each of the parties. This can often be done effectively by using a matrix like the one shown in [Figure 8.23](#).

This type of matrix is best presented to the stakeholders in a meeting, because different people looking at the matrix will have different questions and different assumptions. If the reviewers analyze the matrix unilaterally, they might make a decision

that may be difficult to change later. If the information is presented in a meeting, the decision makers can see other reactions to the information, and any questions and concerns can be dealt with in real time. Similar to what has been discussed earlier, decisions are often better when they are made through an interactive review process.

Figure 8.23 shows that there is a certain flow to how the information should be presented. Start with the criteria for the selection and review the process of how the requirements were weighted. If there is no agreement that these are the right criteria and that the weighting accurately reflects the stakeholders' view, then the rest of the evaluation will be unreliable. After this is reviewed, discuss the alternatives, including any that may already have been rejected, with a rationale for the rejection. Then finally review how each alternative matches up against the selection criteria. By this time, the stakeholders should have a clear picture of the pros and cons of each alternative and should be able to make a decision. A frequent concern with this approach is that the customer wants to change the criteria and the weights of the criteria until the matrix shows the solution that the customer likes. Do not be too concerned about this. What will often happen is that, intuitively, the customer feels that there is something wrong with the selection. And in all honesty, despite the best effort of all involved, this process is subjective in nature. By allowing the discussion and the changing of weights, the process helps by adding structure to the discussion. As with so many tools discussed in this book, it is not the tool that is important, it is the discussion generated by the tool. The end result will still be subjective, but there has been solid discussion and increased understanding of the problem and the solution. This is one of the business analyst's most important and most difficult job functions: to generate and facilitate communication.

8.4 Summary

Gathering requirements without creating a model is similar to building a house without a blueprint. It severely limits your chance of developing the right product. The choice of modeling technique is less important than consistent use of that technique. Once the whole organization is trained on a certain technique, the value gained from the technique will greatly increase. Frequent changes of modeling techniques will cause confusion and often produce poor results. Combining different techniques is OK, but be aware of duplication of information. Train all stakeholders on the modeling techniques before using them in requirements sessions. Keep in mind that modeling is primarily a communications technique. If the customer does not relate to it or if it does not help the developer work, then the wrong technique is being used. There may be a need to use different techniques for the customer and the developers. That goes back to the business analyst's role as a translator and liaison.

Activity

1. Review the case study in [Chapter 10](#). Document a requirement for each of the following categories:

- External
- Business—strategic
- Business—tactical
- Business—operational
- User
- Quality of service
- Assumption
- Constraint
- Implementation

When documenting these requirements, focus on making them specific, unambiguous, and verifiable. Write the requirements so that they will be meaningful to someone reading them 3 months from now without having been involved with the development effort. If you do that, many misunderstandings will be avoided in the future.

2. Review the case study section called *Interview with Store Buyer*. If you are using modeling techniques different from the ones used in the exercise, it may be productive to model this business area using those techniques as well.
 - a. On the basis of the interview, what use cases are needed for this business area? Identify three to five use cases. Select one of those use cases and fill out the use case template.
 - b. Identify the key entities that the buyer's business area deals with, identify attributes and relationships for those entities, and outline an ERD.
 - c. Write a requirements statement for the *Placing special order* requirement. Make the statement specific, measurable, and unambiguous.

SOLUTION EVALUATION

Facts are the enemy of truth.

Don Quixote, in *Man of La Mancha*

This chapter looks at the business analyst's involvement with the project after the requirements have been gathered and agreed upon. The level of involvement from the business analyst will vary greatly from organization to organization; however, typical activities that the business analyst may be involved with include selecting the solution, technical proposal analysis, training, impact analysis, and communication.

9.1 Objectives

- Match a solution to the requirements.
- Make sure that the solution is usable.
- Review different types of tests and the role of the business analyst during the testing activities.
- Communicate the impact of the solution to the organization.
- Support the implementation of the solution.

9.2 Overview

The business analyst is responsible for the requirements of the solution, not the solution itself. This means that the business analyst has the responsibility for tracing the requirements through to the product. Obviously, this must be done together with the systems analyst and the quality assurance people; nevertheless, it is the business analyst who should have the conversation with the customer on how the requirements have been fulfilled by the solution, what the impact will be on the business environment, and what needs to happen to prepare the users for the implementation. It is also fairly common that there are requirements that cannot easily be implemented or that, if changed, would be significantly easier to implement. The business analyst should be involved with the requirements negotiations that inevitably take place on any complex project.

The business analyst is responsible for the requirements of the solution, not the solution itself.

9.3 IIBA Tasks

The International Institute of Business Analysis™ (IIBA™) has identified five tasks that fall under the topic of solution assessment and validation:

1. Measure solution performance. Evaluating solution performance is mostly done after the implementation of the solution. It includes reviewing how the solution is being used, how the users have potentially modified the solution, and what value the solution actually brings. The performance of the solution should be evaluated both quantitatively, using techniques such as ongoing cost, opportunity cost, and return on investment, and qualitatively, with techniques to assess stakeholder satisfaction.
2. Analyze performance measures. The business analyst must be able to understand the value expected from an initiative and to link measurements to that value in order to ensure a successful project.
3. Assess solution limitations. There is often a discrepancy between what the customer really wants and what is cost-effective to deliver. Especially when we are delivering commercial off-the-shelf software, but also in custom development.
4. Assess enterprise limitations. Even if the solution is perfect, there are often enterprise limitations that will impact value realization. Items such as competency, processes, and resources are good examples.
5. Recommend actions to increase solution value. What can be done to increase the value of the initiative over time? Perhaps with increased resources and enhanced technology, there can be a plan for the future.

9.4 Matching the Solution to the Needs of the Customer

Once the solution has been selected, there must be a mapping of the solution back to the requirements. The main concern is to identify requirements that have not been met by the solution or have only partially been met. All of these *missed* requirements must be evaluated and brought back to the customer, and a decision must be made on what to do about them. Maybe the package can be customized, maybe the customer can live without that requirement, or maybe there is a different system or business process that can provide this capability. It is rarely realistic to expect that all of the customer's requirements will be met, and in most cases the customer understands this as well. Customers expect that some things will happen, some things will be deferred, and some things will be dropped. They may not act like they understand this, but often it is necessary to make trade-offs with time, cost, or other factors to come up with an acceptable product. An acceptable product does not always mean meeting all the customer's requirements. This is a time when the negotiation skills of the business analyst, the project manager, and other stakeholders are put to the test.

There is also a need to evaluate what excess functionality the customer will get from this solution, especially if it is a package solution. This part of the evaluation is

easy to forget because there is a tendency to think that excess functionality is at best positive, giving customers more than they asked for, or at worst is neutral because the customers will not use it anyway. In reality, though, excess functionality can be a real negative. It may be duplicating functionality already existing elsewhere, which could confuse the customers or cause duplication of work. It may also prove to be more expensive to support and maintain this extra functionality, which typically is not included in training and testing plans. So carefully evaluate this side of the equation as well, and decide how to approach and handle the extras.

9.5 Support Testing and Quality Assurance

Quality assurance is more than just testing; it also includes making sure that the process is being followed and that a well-thought-out business strategy is developed. Many times, a test plan is created without considering the entire strategy. Just writing test cases and running through them during the project's testing cycle may be catching the defects, but it will probably not create great customer confidence. The purpose of the test strategy is to look at what, when, how much, where, and who. It should be a formal document, which is signed off by the customer or sponsor or both. [Figure 9.1](#) shows an example of a test strategy document for the Prescription Interaction Project.

Section	Example
Overview	This document outlines the testing approach for the user acceptance test for the Prescription Interaction Project.
Intended audience	Business analysts Developers Users
Operational environment	The system test will involve pharmacy, accounting, and customer support
Test environment	The corporate training facility for pharmacists will be used for the test
Testing resources	Three preselected pharmacists Business analyst
Tolerance to defects	Low, there is a high liability risk if the product does not catch dangerous drug interactions
Test types	Stress test Usability test Functionality test
Critical test success factors	All potential drug interactions caught in test cases
Sign-off	_____

Figure 9.1 Test strategy document.

The key sections to include are

- Overview—A high-level description of the project and the purpose of this document.
- Intended audience—Who will receive this document and what should they do with it?
- Operational environment—Which organizations will be impacted by this product and where are they located?
- Test environment—What setup will be used for the test?
- Testing resources—Who will perform the test and what equipment will they need?
- Tolerance to defects—What will happen if there are errors after testing is done?
- Test types—What special situations will be tested for?
- Critical test success factors—What will it take to get sign-off?
- Sign-off—Signature of decision maker.

The test plan for a project should be initiated as soon as the requirements are captured and documented. In many organizations, it is the role of the quality assurance function to create and execute test plans, but it is also common that some or all of the responsibility falls on the business analyst. The testing topics that will be covered in this section are

- Validation versus verification
- Planning for the test
- Types of tests (black box, regression, usability, stress)
- Evaluating customer satisfaction
- Acceptance test

9.5.1 Validation versus Verification

The topics of validation and verification was earlier described in [Section 8.3.2](#).

9.5.2 Planning for the Test

A test plan has many components. It should reference the test strategy document created earlier (who, what, when, etc.), it should outline test types (user test, stress test, etc.), and it should have test cases. While the test strategy document is for organizational commitment, the test plan is a more detailed document outlining the testing process for the project team. The first portion of the plan should be developed during the analysis phase of the project, often by the business analysts or least with significant input from them. Although a detailed test case with well-defined test data may not be possible until the system design is done, the vast majority of

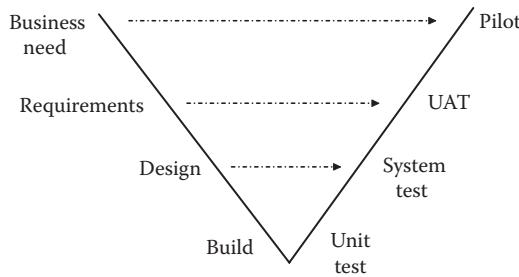


Figure 9.2 V testing cycle.

test cases should be identified, at least at a summary level, during the analysis phase of the project. In general, a good testing approach follows the V cycle of testing, as shown in [Figure 9.2](#).

The strength of this visual is that it shows that you start developing the test plans for Business Need, Requirements, Design, and Build during those phases, and then execute the actual test cases during the testing phase. It encourages the business analyst to start developing the test plans early. Obviously, the business analyst is not responsible for all of this. Design and build testing should be done by the systems analyst and the developers, or possibly by quality assurance if that is the way the organization is structured. Regardless, the business analyst should be aware of all the testing going on because any potential errors or questions that arise during the testing may need to be discussed with the customer, and this often results in new requirements being discovered. Typical sections of a test plan include

- Introduction/overview, stating what this document contains, the intended audience, and a brief description of the project.
- Testing types covered by this document, which may be all types of tests or those focused specifically on acceptance, integration, or usability testing, etc. These types are covered in more detail in [Section 9.5.3](#). Note that there will be subdocuments outlining the detailed plan for each type of test.
- Roles and responsibilities. Who will be involved with the testing? Who is responsible for what portion of the testing effort?
- Authority. Who can sign off on the test plan and who can approve each individual test result?
- Test scenarios, which describe the tests that need to happen without going into the details of what data to input and what data to expect out. Each test scenario will lead to multiple test cases, as seen in [Figure 9.3](#).

The test scenario describes what should be tested: “Customer attempting to fill a prescription for a drug that has a negative interaction history with another drug that the patient is taking.” This scenario may have test cases involving incorrect data, patient no longer taking the previous drug, patient overriding the warning, etc. The business analyst should identify at least one test scenario for each requirement. It is

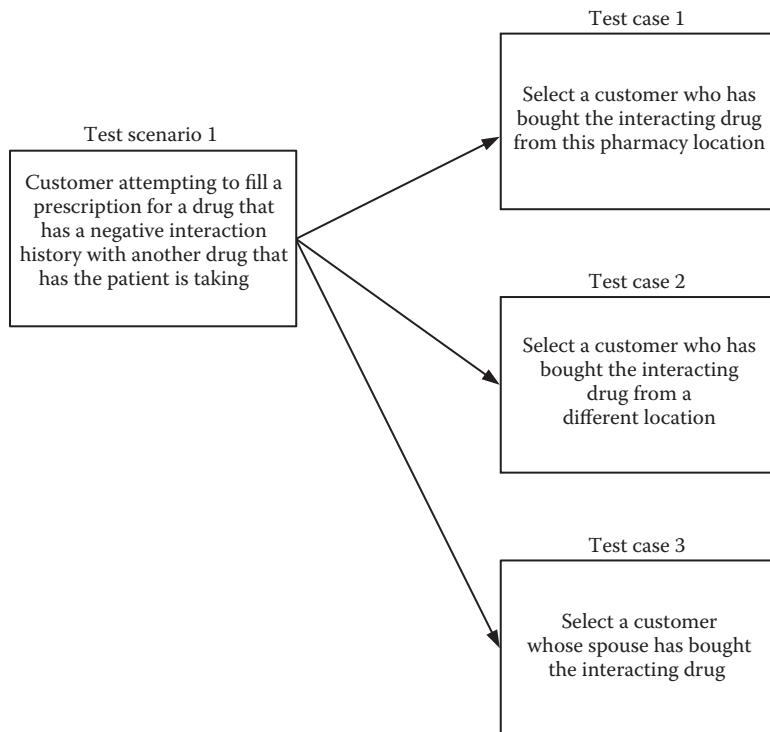


Figure 9.3 Test scenarios and test cases.

often good to consider one successful scenario and one unsuccessful scenario. Some of the details of each test case may need to wait until the design phase of the project.

- **Test cases:** The test cases take the testing scenario and add input data and expected output data. There will normally be more than one test case for each scenario, at least one leading to a successful completion and normally many leading to different error conditions.
- **Test schedule:** The test activities should be included in the project plan from the beginning of the project. Sufficient time must be planned for all the different types of testing as well as time for recovery from errors found during the test. It is common to see a project schedule containing a unit test, system test, integration test, and user acceptance test, followed by implementation, without time set aside between the tests to evaluate and act on the test results. There is a natural hesitation to do this because it implies that errors are anticipated. However, not allowing time for this creates the risk that the project will not be implemented on time. To determine how much time should be set aside, the best guideline is to look at history. What is the normal level of defects found for each test and how much time is normally spent correcting them? This will at least provide rough guidance for the plan.
- **Testing environment:** The most difficult part about performing a test is to simulate of the *live* environment realistically. Creating a setup with system

interfaces, realistic historical data, and valid user inputs can be very time consuming and, even in the best of circumstances, will be a compromise. Some organizations have a *model office* environment that allows them to set up an adequate environment. Other organizations will need to spend time creating data, simulating interfaces, and creating user inputs. This effort must be accounted for in the project plan and should start early in the project.

- Test reporting: This describes how the test results will be reported, the report format, and who needs to see the reports.
- Assumptions, issues, and risks: What are the open items at this time? What risks are discovered using the selected testing approach?
- Version control: This is a document that will go through numerous changes during the project cycle. It is important that every affected stakeholder has access to the most recent version.

9.5.3 Types of Tests

There are many types of tests done and even more names for them, such as regression, stress, and usability. Not all of them fall within the domain of the business analyst, but the analyst should at least be familiar with them. There can often be synergy between the different tests, especially in the area of setting up the test environment and identifying testing scenarios. Be aware, though, that using the same testing environment and test scenarios for all the tests increases the chance of overlooking certain conditions. It may be beneficial to have different environments for the integration test and for the system test, for example. By comparing the results from both, the value of each test is enhanced.

The business analyst should be familiar with the following types of testing:

- Unit test: This test, typically done by the development team, makes sure that the system was programmed according to specifications. It is recommended that the developers of the actual unit being tested do not test their own work, to avoid a situation where developers test only what they already know is working. Rather, have a peer on the team do the testing.
- System test: The primary purpose of this test is to evaluate whether the design is implemented correctly. This can be done in layers, first testing subsystems and then gradually going to a higher level of testing. Does the flow reflect the design? Is the right data available? Although this is normally the responsibility of the systems analyst (or quality assurance analyst), there is a real benefit for the business analyst to be involved. It may point out surprises in how a requirement was implemented and will give a head start in addressing potential issues with the users.
- Integration test: This is often a very difficult test to set up and often needs involvement from other organizations. The purpose is to make sure that the

system being developed correctly communicates with other systems and organizations. The business analysts and the systems analysts will normally need to work closely together to plan and execute this test, not only within the project team, but also with other organizations that may not even be interested in this project. There is a lot of negotiation and relationship building needed to successfully complete an integration test. Start early with ongoing communication with both the management and the support team for the areas with which the project will interface.

- Regression test: This test is really looking to answer the question, “Did something unrelated break because this system was developed?” Although this is part of the business analyst’s job, it is not done in the analysis phase. This is not a test based on requirements, but rather a test of the areas supposedly unaffected by the requirements. Some organizations have automated testing tools that will run predefined testing scripts to see if a system works. This is a great way to do regression testing because it already has the test cases built in and requires a minimum amount of customer involvement. If this option does not exist, then at a minimum identify the high-visibility functions within the business and schedule a user to run through a few key transactions. Even though this is not guaranteed to stop all errors, it gives a reasonable chance to identify any glaring mistakes.
- Usability test: A system can be meeting all functional requirements, but when the user sits down to use it, it is just not user-friendly. Or, although it is performing the right functions, it is not practical in the user’s work environment. The business analyst is often the person primarily responsible for usability testing, both to define it and to work on executing it. Things to look for in usability testing are
 - Is the user interface consistent with other systems that the user uses on a daily basis? If the customers are used to a certain look and feel of their application, based on it being a commercial off-the-shelf product or based on standardized designs within the organization, it is important that this system has the same look and feel.
 - Are the equipment and user interface practical in the work environment? As an example, voice recognition may be great for a phone answering system, but it may not work in a busy or loud office environment. If the system is to be carried around by the user while doing work, such as an inventory system, there are weight and size considerations that may make a system impractical.
 - Is the flow through the system consistent with the way work should be done? It is not enough to test that the functions can be done; the product must also provide the ability to do them in a logical flow, sometimes one that is flexible, based on the user’s situation. If the system does not flow properly, there is a great chance that although the system will pass the test, it will run into problems when a larger number of users start using it in a production environment.

- Stress test: This is often a difficult test to create and execute, but there are tools that will help in some environments. The purpose here is to determine the level of activity on a system at peak times, and then create that scenario to ensure that the system remains viable. Although it is realistic to expect some degradation in performance, it is important to define what an acceptable level of degradation is. Most estimates for peak activity levels are based on analysis of the current situation, so always build in extra capacity to account for unexpected growth. The business analyst should be heavily involved with defining the criteria for the stress test; however, the execution of it is often done by the systems analyst.
- Acceptance test: This is the users' time to verify that everything was implemented on the basis of requirements and that the system is ready for production. The basis for this test should have been developed during the analysis phase, and the business analyst is responsible for being the driving force behind this test. However, the business analyst should not actually do all the testing, but should have the users heavily involved at this stage. If they are not, it will be too easy for them to disassociate themselves from any responsibility for the system. One important goal that the business analyst must have throughout the development process is to have the customer take ownership of the product. The acceptance test is part of that process, and if that is delegated to the analyst, it is likely to cause problems down the road.
- Beta/pilot testing: This refers to a limited production release of a system (or a part of the system) to a small portion of the organization. Although the terms themselves are not used consistently in the industry, the intent of the beta or pilot testing is to allow the customer to use the product for a while to test how well it is functioning. It will also help evaluate any implementation and training issues. One of the great risks with implementing a system is the impact on the user environment. It is a good risk mitigation strategy to do a gradual roll-out of the system to allow time to identify lessons learned and make process improvements. Be aware that there are some downsides because the support staff must support both the old and the new system, and any external interfaces must also be duplicated. However, for a large, complex project, it can still be a good idea to perform these types of tests.

9.5.4 Evaluating Customer Satisfaction

Beyond the testing itself, there should also be a long-term evaluation of how well the product has been received by the customer. The business analyst should plan on conducting surveys to verify this. One can be done shortly after implementation (1–3 months later) to evaluate training, implementation, support, and process issues. A second one can be done with a long-term view (12–18 months) to

determine how well the product actually solved the business problem, basically asking the question, “If we could start over, would we do it again?” or “Was it worth the effort?” Make sure to request feedback from the paying customers, internal management, and of course, the users. The results of the evaluation, which become a lessons learned document, can be used for future process improvements as well as for reaffirming with the customer that this was a successful project. “Here’s what the problem was, here’s how it was fixed, and here’s the final result.” This should be part of the sales job for the development organization. The best time to ask for more business or to promote an organization is after it has successfully implemented a new product.

9.6 Implementing and Supporting the Solution

Often when the original project plan is created, there is a lack of detail on implementation and support issues. This is partly because these activities often occur after the project manager is officially done with the project, and also because they seem so distant on the horizon. There are a number of factors to consider in this area, such as

- Conversions—The conversion effort should start in the analysis phase when the as-is and to-be data models are being created. On the basis of these models, a conversion process must be developed to decide what data gets converted and how it is mapped to the new system. This process is not trivial and must involve both the business analyst and the customer. Some of the conversion may be automated (we hope most of it), but some of it (especially exceptions) may need to be manually reviewed and decided on.
- Business transitions—What needs to change on the business side to take advantage of the new system? This can include user training on business processes, process changes, changed job descriptions, and elimination of job functions as well as creation of new ones. While this must be led by the business, the business analyst will often be involved.
- Communication of the implementation plan to the users—It may seem obvious, but sometimes the users do not know how a system will be implemented, what functionality goes away with the old system, and what new functionality is added. This is extra important for those stakeholders who may not have been closely involved with the development effort. This step may include training on the system and providing documentation.
- Communication with other affected organizations—It is rare that a new product impacts only the organization for which it was developed. Most of the time, there are interfaces, both from a systems view and from an organization’s view, and those interfaces must still be working on implementation day. Although it is mostly the system analyst who will verify the pure system interfaces, the business analyst should communicate with the affected

organizations and make them aware of what may change when a new system is being implemented and to watch out for any abnormalities. It can be difficult enough for an outside organization to be impacted, but if they are not told ahead of time about any potential impact, it will be an even more frustrating experience for them.

9.7 Summary

Although the business analysts do not own the design, development, or implementation of the solution, they are heavily involved with it. Make sure that the roles and responsibilities reflect this, and have the business analysts plan their activities not only for the analysis phase, but also for all their involvement on the project. A good business analyst is able to discuss all aspects of the product development with the customer. Although the project manager owns the project areas of cost, schedule, and scope, the business analyst owns the product aspect, the ability to trace the product in use by the customer back through the testing, the design, the requirements, and ultimately all the way back to the business problem that initiated the need for the product. This takes a set of skills and characteristics that are difficult to find but, when found, are invaluable to the business.

Activity

1. Describe three parts of the Swede-Mart case study (see [Chapter 10](#)) where the customer is likely to have a very low risk tolerance.
2. The following requirement was documented for the case study: The system must verify that the customer has sufficient credit to place an order if the order is above \$1000 and if the customer does not have an existing line of credit with us. Write two test scenarios that would help verify this requirement.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

SWEDE-MART CASE STUDY

The following description of a fictitious retailer called Swede-Mart is to be used to complete the Activity section at the end of each chapter, applying the concepts discussed therein. Sample solutions for each of the activities are given in [Chapter 12](#).

10.1 Introduction

Swede-Mart is a rapidly expanding clothing store headquartered in Gothenburg, Sweden, with 300 stores around the globe, 200 of which are in Europe. The retailer targets affluent families who want fashionable clothing, but who are also concerned about price.

Swede-Mart has a 3-year plan to greatly expand its presence in Asia and North America. The goal is to become the largest clothing store in the world in the market of upper-middle-class, young families, as measured by number of stores and revenue.

Swede-Mart offers high-quality products, competitive prices, and a trendy image. Holding on to this image will be key as the company expands. The target market is very finicky and will jump ship quickly if the company's image does not match their lifestyle.

10.2 Strategy

Swede-Mart has recently gone through a business planning effort and formulated its mission and vision; however, a strengths–weaknesses–opportunities–threats (SWOT) analysis has not yet been done.

The initial focus of the strategic plan is ensuring customer retention and growth through inventory management, which will allow Swede-Mart to have the right products available at the right location, at the right time, and for the right price. This has been a major part of the company's marketing campaign. This means setting up regional distribution centers that can ship efficiently to local stores. The company will need to make sure that it can execute this as well at a global level as it has in Sweden.

The goal is to have less than 1 percent of an advertised product out of stock over the duration of advertising. Year-round inventory should turn eight times per year; currently, it is only six times.

Successful improvement of the inventory management system will allow Swede-Mart to increase its rate of growth across the globe, and will also provide cash flow for acquisitions and store growth.

10.3 Industry Background

The retail clothing industry has always been a high-risk industry with frequent turmoil. It is a very competitive market. Anyone who has a successful product line can make good profits, but missteps will be paid for dearly through lower margins and unused inventory. In the case of Swede-Mart, some of its products can also be bought in many other stores, although some brands are unique to the chain. The discount market for clothing operates on very thin margins and counts on high volumes and unsold merchandise from other companies. Swede-Mart is considered a mid-range clothing store, competing with the likes of Gap and H&M.

The key functions of the business are purchasing products from designers, sourcing the manufacturing of the products, managing the inventory process, and marketing and selling the products to the customer.

The selection of the right products based on recent trends and finding the right price point are keys to success in this industry. Competition is fierce and each company must find an image that helps it to sell. Swede-Mart's image is of a retailer that sells clothes that are fashionable and available in a fun and well-laid-out store. That puts the company's success in the hands of the buyers and the inventory management system.

The largest costs are space, inventory, and staff. Not only must the product be there at the right time, there must also be enough staff there to support the sales. Swede-Mart is famous for its friendly but not overbearing staff.

10.4 Project Background

Joe Jones is the sponsor of this initiative to enhance the inventory and order processing functions for the distribution centers. Joe has been with Swede-Mart for 20 years and has been in charge of the distribution centers for the last 3 years. He is widely assumed to be the next president of the company.

Joe is known to want frequent, informal communications. He is OK with bad news, but he wants it early and he wants ideas on how to fix it. In his general directions to the team, he asks not to be surprised. He is always willing to get involved and help, but he is not a micro-manager, so he will assume that things are OK unless he is told otherwise.

10.5 Distribution Center/Inventory Operations

The initial focus from the strategic plan will concentrate on reviewing the inventory-management and order-taking areas. Distribution centers are organized in regions,

with each region containing one or more distribution centers. Each distribution center has up to twenty-five stores that it supports. This may need to be adjusted as the business grows. Optimizing the inventory levels between the distribution centers and quickly recognizing increased demand will be drivers of success.

10.6 Product Lines

There are three main product lines:

- Seasonal—These products will change every season, but they are pretty much standard for the season (shorts, winter coats, etc.). There are good statistics for use, but any unusual weather patterns will cause huge fluctuations.
- Special fashion—These products change on a nonseasonal basis and are often related to a special event or new fashion, such as for a sporting event, New Year's Eve, or a brand-new fashion trend. These are short-term events, with not much residual value of the product. Leftover products go to clearance.
- Year-round—These products are always in stock, always in demand, for example, socks, underwear, and suits. Their sales volumes are predictable, and competition is mainly focused on pricing. The inventory for these items turns over frequently.

Products are identified by a product code. Each manufacturer has unique product codes, so Swede-Mart must use its own product codes internally. Each store has a buyer who can order from the Swede-Mart product catalog. Not all products will be in stock during all seasons. The distribution center is responsible for determining stock levels. Some products can be special ordered, with a longer lead time. This can be done for products not stored by the distribution centers (typically high-priced items), or for items out of season. Naturally, the delivery time will be much longer for these types of items.

Although many items are purchased directly from the manufacturer, often there is a third party involved, especially for low-volume products.

10.7 Purchasing

Buyers are assigned to product lines and to vendors. Where a vendor is providing multiple product lines, there is a primary buyer who is responsible for the price negotiations and the general vendor relationship. Each buyer buys for all the distribution centers. Most of the buyers are at the headquarters in Sweden, but some are located at distribution centers around the world. This is likely to increase with the global expansion over the next few years.

The buyer will get sales and predicted demands from the stores and distribution centers, and then do a sales forecast based on that data. The buyer will negotiate prices with the vendors and look out for special discounts or sales from the vendors to take advantage of opportunities as they arise.

Stocked items will have a reorder level. This level is set for each stocked product, for each store, and for each distribution center. When a product reaches a reorder level, an automatic order will be generated by the system. This reorder level is updated by the distribution center as order patterns change.

10.8 Receiving

Vendor shipments typically will be shipped to the distribution centers. Occasionally, special orders may be shipped directly to the stores. Receiving will match the shipments to purchase orders to make sure the shipment is correct and that it has arrived at the right location. After that, items that are to be stocked in the distribution center are entered into inventory and stored in their bins.

Products that are to be reshipped directly to the stores get sent over to shipping and combined with any other shipments being sent. On average, the distribution centers send one large shipment per week to each store.

The term *bin* in the distribution center refers to a specific location where a product is stored. It could be a shelf, a bin, or a corner of the warehouse. Each bin is described by its type and size and also has information on location (floor number, part of building). Any changes to a bin location must be immediately reflected to make sure that the warehouse will operate efficiently. A product may be stored in multiple bins and, in some cases, multiple products will share one bin.

10.9 Accounts Payable

Vendor invoices are matched to a purchase order. The purchase order is updated by *Receiving* when a shipment is received, and no invoice will be approved for payment until the shipment has arrived and been approved. The invoice price is compared with the purchase order price, and the payment terms are compared as well. If there are discrepancies, those are forwarded to the buyer for resolution. Once all of this is approved, the invoice is marked as payable.

The payment date will be based on the terms from the invoice. If there are discounts offered for early payments or penalties for late payments, then those discounts or penalties are annualized and compared to the organization's cost of funds to determine whether there should be an early or late payment.

There must be an ability to find previously paid invoices and to know their status, including knowing what the cost of funds were at the time of payment.

10.10 Order Processing and Shipping

When the stores place orders with the distribution center, products can be shipped to the store for retail sales. Typically, orders are placed once each week and then shipped the following week. A store can place an emergency order, which will sometimes

happen when there has been an unusual weather pattern or a new product is more successful than expected. This type of order can be placed at any time, and is typically shipped out the same day if received before 3:00 p.m. local time. Depending on the cost of shipping and any other shipments going out, the shipment may be delayed to make it financially viable. The store will always be involved in that decision process and will be responsible for all shipping charges.

When an order is received from a store, the distribution center checks the inventory levels to see if there is enough inventory. All items that are in stock are confirmed, and a notice is sent back to the store with the expected shipping date. If there is not enough quantity in stock, then a partial confirmation will be made. If no stock is available, that line item will be cancelled. The store will have to try to reorder the following week if the product is still wanted. If there is a special need to get a product earlier, then the store needs to place a special order.

If the store order is for a product that is not carried in inventory, then a special order to the supplier is placed. Once a delivery date is received from the supplier, this information is passed on to the store. These products are typically shipped directly to the store. All store orders will typically be from their primary distribution center. On rare occasions, a store can place an order with a different distribution center; however, such orders will be handled manually.

Stores can continuously check on the status of an order and will frequently call before placing an order to find out about the inventory situation. Often, they will ask for similar products if the one that they were interested in ordering is not available. The distribution center will invoice the store for the orders shipped, as a way of auditing financial discipline.

10.11 Reporting

There are a number of reports that must be created for this business area. Some of the key reports are

- Invoice report, which shows payable invoices and payment due date.
- Inventory report, which is a complete list of products and quantities.
- Store order report, which shows all stores for a distribution center and what they have ordered. This report must be available in multiple sorting sequences.

All reports will be in English, the corporate language.

10.12 Summary

The new order management system that will be created for Swede-Mart must be state of the art and must be able to support the aggressive growth goals of the company. Although a timely solution is needed and cost is important, this project is really focused on doing the right thing. The solution must be flexible, accurate, and very

responsive. In recent decades, there have been many retail companies that have failed because they were not able to get the right product to the right place in the right time frame. Swede-Mart is determined not to be another one of those companies.

10.13 Interview with Store Buyer

I will use the system to check on the status of system-generated orders. Sometimes, there may be an issue with them and I will have to call the distribution center to make changes. I normally check if there were any line items that were not in inventory and if there are any expected delays in ship dates. I also need to be able to look up a product and see the inventory levels in our store, at the primary distribution center, and at other distribution centers. This should be displayed so I see the centers in sequence of expected shipping times.

I must be able to place special orders; this normally happens when demand increases unexpectedly or when the system-generated order is not correct. A special order is also placed for products that we carry in our catalogue but are not available in our stores. These items are typically high priced and have a very low sales volume.

CBAP TEST QUESTIONS

- Q1. The finance department of Shorlander Inc. is looking for a new business analyst. They are planning to move toward a package solution for all of their financial applications. What skills would be most important to look for in the candidates interviewed?
- IT knowledge and presentation skills
 - IT knowledge and business understanding
 - Leadership skills and problem solving
 - Business understanding and problem solving
- Q2. Which of the following would qualify as a requirement?
- A capability needed to solve a problem
 - A goal that the project should achieve
 - A solution presented by the technical team
 - A result reached in a system
- Q3. The IIBA talks about the need for the business analyst to have systems thinking. What does that mean?
- Understanding capabilities of the current and future IT system
 - Understanding how people, processes, and technology interact
 - Being able to understand how customers and government impact the business
 - A step-by-step approach to problem solving
- Q4. During a customer interview, the analyst gets challenged by the customer in regards to the purpose of the interview. The customer is not seeing the value of the session and wants to get an understanding of why he is there. One of the issues is that he “doesn’t even know what a requirement is.” He has asked for a number of things that have been rejected and the heat is increasing. Which of the following would be the best statement to start with?
- A requirement tells what you want the system to do, not how it would do it. The *how* will be determined later.
 - A requirement is something that is needed in order to solve a problem. It should be focused on meeting an objective.
 - A requirement is a fully detailed statement that is verifiable at the end of the project.
 - A requirement is something that has been approved by the sponsor of the project and that is funded.

- Q5.** The project manager overheard you talk about the scope with the customer. She calls you into her office and appears upset. She explains to you that the scope is part of her domain and that she wants you to stick to gathering requirements in the future. What would be your response?
- a. You were only discussing the scope of the requirements-gathering effort, which is your responsibility. Once you are done determining that, you will let the project manager know the results.
 - b. Since you do not report directly to the project manager, it would be more appropriate to bring this up with your manager.
 - c. You are determining the solution scope, which is really an input into the project management processes. There is no point in the project manager starting to worry about the project scope before you know what the customer wants.
 - d. What you had been talking to the customer about was the solution scope; in other words, the product description. You agree that you should not discuss project scope directly with the customer, unless you and the project manager have agreed on that ahead of time.
- Q6.** At the post mortem of a recent project, the customer complains about some missing functionality in the product delivered to them. When you point out that those features were never part of the requirements document, the customer gets irate and lets you know that anyone with any business sense should have understood that these features were essential. What went wrong?
- a. The customer is focusing on what he wants rather than on what he needs.
 - b. The business analyst needs more business training.
 - c. The business analyst should have done a better job documenting assumptions and expectations.
 - d. The customer did not build consensus with other stakeholders in regards to their requirements.
- Q7.** In a meeting with end users, they explain their goals for using the system. What type of requirements are these?
- a. Enterprise requirements
 - b. Business requirements
 - c. Stakeholder requirements
 - d. Functional requirements
- Q8.** You are leading a JAD session with a cross section of stakeholders to determine nonfunctional requirements, and thus far you have just generated a lot of confusion. Some stakeholders think that nonfunctional means broken; others think it means low-priority items. Which of these descriptions best describes a nonfunctional requirement?
- a. Nonfunctional requirements describe the characteristics of a system and the environment it needs to operate in.

- b. Nonfunctional requirements describe the requirements that are important for the developers to know the technical constraints of the system. They are typically not of primary concern to the business.
- c. Nonfunctional requirements describe the qualities of the system. They can help make the system easier to use, but they are secondary to functional requirements.
- d. Nonfunctional requirements describe the system, whereas functional requirements describe the business.

Q9. Your organization is looking to hire a new business analyst. You are assigned to write the job description for job posting. Which one of these job postings would be the best?

- a. Business analyst: Create and validate requirements documents for the customer organization and ensure that those requirements are implemented correctly by the developing organization.
- b. Business analyst: Facilitate requirements-gathering sessions to build consensus regarding project requirements.
- c. Business analyst: Liaison between the business and the developer.
- d. Business analyst: Interface with all relevant stakeholders to understand the policies and goals of the business. Work with others to help recommend solutions that will satisfy those goals.

Q10. While meeting with the development team of the new payroll system for the A.J. Corporation, they inform you that many of the users have very limited computer knowledge. Because of that, the customer has asked the IT organization to perform some basic computer training when the system is implemented. The team leader of the development team implies that she thinks that this would be your responsibility. It does not sound right to you, but you decide to go back and do some research. Which of these statements would be in line with the *BABOK*?

- a. This is dealing with the solution implementation and should fall under the development team's responsibility.
- b. This is an implementation requirement and needs to be submitted as a change control and approved by the change control board.
- c. This has nothing to do with this project and its goals. It is clearly scope creep and it should be rejected.
- d. This is essential in making the customer successful. The business analyst must take care of this.

Q11. Which of the following would not be considered *noise* when you are communicating?

- a. Distance
- b. Language
- c. Culture
- d. Message

Q12. Your deadline is very tight for capturing requirements and this project is key for your organization. There has been some conflict between you and one of the people on your team, who resists following accepted standards. Which conflict resolution technique should you choose?

- a. Forcing
- b. Smoothing
- c. Compromise
- d. Avoiding

Q13. What does a stakeholder requirement show?

- a. All requirements that are given by stakeholders outside of the project domain
- b. A stakeholder's need when interfacing with the solution
- c. How the solution should behave for a certain stakeholder
- d. Which stakeholders must sign off on the requirements document

Q14. Maslow's hierarchy of needs can be useful for a business analyst. How?

- a. It can organize business needs.
- b. It can evaluate the needs of the end users.
- c. It can be used when designing workshops to make sure the participants are up to the task.
- d. This is more for the project manager.

Q15. Which of the following statements is true about the business analyst and the domain subject-matter expert?

- a. These are different names for the same things.
- b. Domain subject-matter expert is a subset of the business analyst role.
- c. Domain subject-matter expert is sometimes a role played by the business analyst.
- d. The domain subject-matter expert role must be performed by someone other than the business analyst.

Q16. The business analyst typically has extensive communication with the customer and users. That is also true for the project manager. Which of the following statements is true?

- a. The business analyst is responsible for the solution scope.
- b. The business analyst is responsible for capturing the solution need.
- c. The business analyst is responsible for the project scope.
- d. The business analyst is responsible for capturing the project need.

Q17. When a business analyst evaluates potential solutions, he or she must be aware of sunk cost. What does this mean?

- a. That a certain amount of money must be spent on a project before we know if it is viable.
- b. That we must consider how much we already have invested before we discard a solution.

- c. That there is future cost and there is sunk cost. We can only control future cost.
- d. That the fact that we have spent money on a project does not mean it should continue.

Q18. "The solution must allow the customer to enter invoices with payment terms." What type of requirement is this?

- a. A functional requirement
- b. A stakeholder requirement
- c. A transition requirement
- d. A business requirement

Q19. "All invoices in process at time of implementation must be moved over to the new solution." What type of requirement is this?

- a. A functional requirement
- b. A stakeholder requirement
- c. A transition requirement
- d. A business requirement

Q20. "The audit department must be able to track late payments on invoices." What type of requirement is this?

- a. A functional requirement
- b. A stakeholder requirement
- c. A transition requirement
- d. A business requirement

Q21. When the business analyst takes on the role of a facilitator, it is important that he or she is

- a. Neutral
- b. Knowledgeable about the topic being discussed
- c. A good speaker
- d. Able to explain the issues at hand

Q22. When working on the tasks in the *BABOK*, a business analyst

- a. Should perform the tasks in sequence
- b. Must perform the tasks in sequence
- c. May perform the tasks in any sequence
- d. Must perform the tasks in the sequence outlined in the *BABOK*

Q23. When the business analyst is also the project manager, care should be given to

- a. Balancing solution scope and project scope
- b. Limit the project manager's authority
- c. Have an independent review of all decisions
- d. Let the customer know about the potential of conflicting interest

Q24. You are working on a new development project and the customer is getting ready for a requirements session. He hands you some old flowcharts in some unknown (to you) notation. What should you do?

- a. Explain to the customer that this technique is not supported by the IIBA, and thus is not best practice.
- b. Evaluate the flowcharts with other stakeholders and see if they meet the needs.
- c. Use the flowcharts; no need to redo what has been done.
- d. Let the developers decide.

Q25. In the business analysis planning and monitoring knowledge area

- a. A project plan for the analysis phase is created.
- b. The tracing of deliverables are done.
- c. Project progress is monitored.
- d. It is determined what activities are needed in order to complete a business analysis effort.

Q26. According to the *BABOK*, a task

- a. Is an activity on the project
- b. Is the lowest level of the business analysis plan
- c. Accomplishes something of value
- d. Is something that should be estimated

Q27. What is true of requirements?

- a. They must come from the customer.
- b. They must have a test plan attached to them.
- c. They may have different states (verified, prioritized, etc.).
- d. They must be stated in the terms of the developer.

Q28. A functional requirement describes a requirement with the view of

- a. The user
- b. The solution
- c. The developer
- d. The business analyst

Q29. What is the relationship between input and task?

- a. A task can start when input is present and complete.
- b. A task can start while input is being developed.
- c. The input will start the task.
- d. Inputs are generated by other business analysis tasks.

Q30. Which of the following is NOT an example of requirements classification?

- a. Internal and external
- b. Business, stakeholder, solution, and transition
- c. Needed or not needed
- d. Business and technical

Q31. Mind mapping can be a useful tool for the business analyst. The business analyst may use it to

- a. Design a solution for the customer's problem
- b. Help structure a brainstorming session
- c. Map the minds of different stakeholders
- d. Create test cases

Q32. The Business Analysis Core Concept Module (BACCM), which has been developed by the IIBA, can be used to

- a. Structure the requirements document in an intuitive requirements taxonomy
- b. Describe the profession and domain of business analysis
- c. Help create traceability between regulatory, business, stakeholder, and solution requirements
- d. Describe the customer's business when an enterprise architecture model is created

Q33. The core concepts of the BACCM are

- a. Data, process, product, people
- b. Requirements, solution, business need, strategy, value
- c. People, product, process
- d. Change, need, solution, stakeholder, value, concept

Q34. It is the business analyst's job to ensure that the decision makers of an initiative have enough information to make good decisions. This is dealt with in the BABOK v3 task of

- a. Plan business analysis information management
- b. Plan stakeholder engagement
- c. Plan business analysis governance
- d. Plan business analysis approach

Q35. You are gathering requirements for the security of a new application. You are identifying the roles of the different users and what level of access to data and functions they should have. This can be documented in a

- a. Roles and permissions matrix
- b. Roles and responsibility chart
- c. Logical data model
- d. Use case

Q36. You have been assigned to a new project in your organization. You will try to find common trends in buying habits among your customers and also potential emergent products. The type of analysis that would work well for this is

- a. Data mining
- b. Strategy model

- c. Enterprise analysis
- d. Logical data modeling

Q37. You are working as a business analyst on a new initiative. The customer has difficulties telling you the requirements and you seem to have communication difficulties. You then create a simple prototype to help the customer understand the concepts being discussed. You have just used the BA competency of

- a. Visual thinking
- b. Conceptual thinking
- c. Problem solving
- d. System thinking

Q38. You are performing a stakeholder analysis and want to assess how involved the different stakeholders will be with the project solution. Which technique would help you document this?

- a. The RACI chart
- b. The onion diagram
- c. The context diagram
- d. The communications plan

Q39. Which of the following techniques would work well for managing business analysis performance?

- a. Root-cause analysis
- b. RACI chart
- c. Earned value
- d. SWOT analysis

Q40. Project New Idea has been going on for 3 months. The requirements were validated 3 weeks ago. However, yesterday the customer told the business analyst that they had some new functional requirements. What is the best action?

- a. If the change is minor, implement it. This project has just started.
- b. Submit the change to the change control board.
- c. Persuade the customer to move these changes into the next release.
- d. Let the customer know that once we have validated requirements, they cannot be changed. Any changes would need to be part of a future endeavor.

Q41. You have just started to gather requirements for a new accounts payable system, standardizing processes among four business units. You had your first meeting yesterday and it was a real disappointment to you. It did not seem like anyone was interested, and no one would speak up or give you any ideas. What may be the best approach to try for the next meeting?

- a. Review the roles and responsibilities again to ensure that everyone knows why they are there.

- b. Invite the sponsor to the next meeting so that everyone understands the importance of the project.
- c. Do an icebreaking activity with the team to get them comfortable with each other.
- d. Do not micro-manage the team. They just need some time to get going.

Q42. You are working on a very important project for an external customer. There is a tight deadline and you are not sure that it can be met. You just found out that the solution your company has sold has some security flaws that could impact the customer. Your manager tells you to keep it quiet. The developers are working on a fix, but it will not be done in time for the deadline. Which option is best?

- a. Explain that since you view this as a very serious breech, you must go to the customer and let them know.
- b. Since this is your manager, you must follow his direction. You will try to identify some workarounds to minimize the problem.
- c. Escalate it past your manager.
- d. Resign. This is illegal and you should not get involved with it.

Q43. You have just started on a new large effort to gather requirements. The project manager has asked you to identify risks for your activities. On which risks should the business analyst primarily focus?

- a. Elicitation risks
- b. Solution risks
- c. Implementation risks
- d. Should always look at all risks

Q44. You are the business analyst working on the requirements-gathering plan for a new project. Who should be your primary interface in the organization for review of the plan?

- a. The sponsor
- b. The project manager
- c. Your management
- d. The developers

Q45. Your new organization focuses on agile development. From a communication view, this tends to mean

- a. Only communicate when needed
- b. There are no rules for communication in agile
- c. Frequent, short, informal communication
- d. Formal approval of all steps

Q46. The organization is starting up a new development effort that will create a new sales system for the company. Stakeholders are all over the world and the

project team will have developers in China, the United States, and Sweden. What development life cycle would be best?

- a. Waterfall
- b. Agile
- c. Iterative
- d. Should be subdivided into agile subprojects

Q47. Stakeholder identification is one of the key tasks for the business analyst. Why is that so important?

- a. To minimize the chance of missed requirements
- b. So that everyone feels a part of the project
- c. So everyone can benefit from the solution
- d. If not involved, they may become detractors from the project

Q48. The IIBA is using organizational process assets as input to many of their processes. What does that mean?

- a. Financial needs of the project
- b. Templates or process
- c. Documentation tools
- d. The current system

Q49. You are starting the requirements-gathering effort for a new project and you are responsible for a team of business analysts. You want to make sure that they all know each other and understand the objectives. How do you accomplish this?

- a. Bring in the sponsor for a project overview.
- b. Have an icebreaker activity.
- c. Create a charter for the requirements phase.
- d. Hold a kickoff meeting.

Q50. The new project you are working on will use workshops to elicit requirements. You are concerned about the customer representatives. What action would you take?

- a. Send the representatives to workshop training.
- b. Do interviews with other key representatives to get a complete picture.
- c. Document what type of persons you need for the session and review with the sponsor.
- d. Do the best with the resources given.

Q51. For the new executive information system, the project manager has selected an approach using a prototype to gather requirements and then gradually refine the prototype. This approach

- a. Will likely increase the cost of the project
- b. Is considered a change-driven approach
- c. May work better in design than during requirements determination
- d. Is the best approach for all projects

Q52. When selecting the methodology to use for a certain project, this technique may help

- a. Decision analysis
- b. Quality control
- c. Data modeling
- d. Brainstorming

Q53. Enterprise architecture is an input into stakeholder analysis. What is enterprise architecture?

- a. Organizational roles and responsibilities
- b. Describes organizational units and interactions with other internal and external units
- c. A design for the to-be organization
- d. An organizational chart

Q54. Which of the following is NOT a technique used for stakeholder analysis?

- a. RACI chart
- b. Brainstorming
- c. Risk analysis
- d. Quality assurance

Q55. Sara is asked to join a holographic website feasibility study. She has just gotten her CBAP certification. What would be a natural reaction for Sara?

- a. A business analyst would normally start after the feasibility study is done.
- b. This may be a good way to learn skills unrelated to those of the business analyst.
- c. Great! Who can I get more information from?
- d. That would be OK, but then she should not be part of the project if it goes into the next phase.

Q56. What role does *lessons learned* play in business analysis planning and monitoring?

- a. None, lessons learned is a closing process.
- b. It should be updated with the new plans.
- c. It can guide the selection of methodology and processes for this initiative.
- d. It can tell us which people not to include in this initiative.

Q57. For a change-driven project, it is much more important that stakeholders are

- a. Competent
- b. Committed to the effort
- c. Knowledgeable about IT solutions
- d. Co-located

Q58. The business analyst must consider the level of formality needed in communication. This type of activity normally requires formal communication.

- a. RFQ/RFI/RFP
- b. Status reviews
- c. Process modeling results
- d. Customer meetings

Q59. Which one of these techniques would be the most common and effective approach to communicate requirements?

- a. Informal meetings
- b. Structured walkthrough
- c. Requirements workshop
- d. Audits

Q60. In planning “the requirements management process,” *complexity, urgency, and source* would be considered

- a. Assumptions
- b. Constraints
- c. Requirements attributes
- d. Risk factors

Q61. The most likely example of performance measurement for business analysis work may be

- a. Frequency of changes to requirements
- b. Number of unit tests performed
- c. Adherence to project schedule
- d. Resolution of risks

Q62. When receiving change requests for a project, it is important that the impact analysis includes this often forgotten piece of information.

- a. Resource required for performing the change
- b. Cost of change
- c. Requestor's name
- d. Impact of not doing the change

Q63. The business analysis communication plan is normally

- a. A formal document
- b. Part of the project communication plan
- c. Signed off by the sponsor
- d. An informal document

Q64. One difference between a plan-driven approach and a change-driven approach is about requirements gathering. Which is true?

- a. In a change-driven approach, the requirements are gathered as early as possible.

- b. In a change-driven approach, the requirements are gathered as late as possible.
- c. In a change-driven approach, the requirements are not documented.
- d. In a change-driven approach, the requirements are prioritized.

Q65. For a large project, business analysis activities may be planned using the rolling wave approach. This means that

- a. You only estimate the things that will be done in the next 30 days.
- b. You plan in 30-day increments and then do a gate review.
- c. You plan near-term work in detail and things further out at a high level.
- d. You plan near-term work in detail.

Q66. When using an onion diagram, the stakeholders in the center

- a. Have the most at stake if the product fails
- b. Are the most involved with the solution
- c. Will give the most requirements
- d. Are the decision makers

Q67. Which process modeling technique is also useful for identifying stakeholders?

- a. Use cases
- b. Entity relationship diagram
- c. Sequence diagram
- d. State diagram

Q68. Complexity of stakeholder interaction may be influenced by

- a. Technical difficulties
- b. Number of interfacing processes and systems
- c. Project manager experience level
- d. Buy-in from the receiving organization

Q69. An example of a regulatory framework that may impact the business analysis work is

- a. TOGAF
- b. Zachman
- c. Six Sigma
- d. Sarbanes-Oxley

Q70. Which statement is true about traceability?

- a. It is not important on smaller projects.
- b. It is always a value-added activity.
- c. It adds significant overhead to the business analysis work.
- d. It should be done by the project manager.

Q71. During a JAD session, two participants get into a heated discussion about the true business problem. As the facilitator, you feel that things

are getting out of control. Which facilitator tool best helps you to deal with this?

- a. The parking lot
- b. Setting of objectives
- c. Taking a break
- d. Establishing the ground rules

Q72. The elicitation technique of interface analysis can be extra challenging because

- a. There is normally a lack of expertise on interfaces.
- b. They are outside the scope of most projects so the project manager is not interested.
- c. It may be difficult to get the people responsible for the interfacing systems to work with the project team.
- d. Interfaces are normally undocumented.

Q73. When eliciting ideas from a large population of users with varied backgrounds, the following technique would work best:

- a. Interface analysis
- b. Focus group
- c. Requirements workshop
- d. Brainstorming

Q74. During a JAD session there seems to be a lot of disagreement between stakeholders from different departments. It is difficult to get to agreement on the wording of requirements and the problems with the current situation. Which of the following would be a good starting point to correct this?

- a. A training course for the participants
- b. Negotiating for more experienced resources
- c. Establishing ground rules for the requirements sessions
- d. Creating a data dictionary and glossary

Q75. You have been asked to lead a requirements prioritization session. The requirements have been collected and documented but the scope is too large. Which techniques could help with this?

- a. MoSCoW
- b. Brainstorming
- c. Root-cause analysis
- d. SWOT analysis

Q76. When writing survey questions for an anonymous survey, one must ensure that

- a. The questions are complete by stating all assumptions clearly.
- b. The survey population is large and diverse.

- c. Only a select number of people have access to information about who completed the survey.
- d. The survey questions are easy to understand.

Q77. One of the main difficulties with using prototypes to elicit requirements is

- a. Technical people not understanding what the customer is looking for.
- b. All work on the prototype gets thrown away when the real solution is developed.
- c. Customers may have unrealistic expectations.
- d. A prototype is often too simple to show business complexity.

Q78. You are the facilitator of a requirements workshop and you sense that the group is in the forming stage. What does this tell you about how the group functions and what leadership style you should use?

- a. The group functions well, with no conflict; use a democratic style.
- b. The group functions well; use an autocratic leadership style.
- c. The group functions poorly; use a hands-off approach.
- d. The group functions poorly; use an autocratic approach.

Q79. Your technical team has recommended that you use storyboarding when gathering requirements for the new web-based ordering system. What makes storyboards good?

- a. They are intuitive for users.
- b. They are easy to implement as systems.
- c. They show the complete business requirement document.
- d. They answer *what* without going into *how*.

Q80. When brainstorming, it is most important to

- a. Select the right people.
- b. Stay at a high level.
- c. Clearly state the problem that is to be brainstormed about.
- d. Organize the ideas.

Q81. When working in an integrated systems environment, where the product being developed is replacing a portion of the existing application(s), the following technique should always be part of the elicitation techniques:

- a. Interface analysis
- b. Focus group
- c. Requirements workshop
- d. Brainstorming

Q82. There are different types of questions to be asked in a requirements interview.

Which type of questions would you normally start with?

- a. Closed ended
- b. Open ended

- c. Validating
- d. Probing

Q83. When gathering the requirements for an upgrade to the existing inventory system, it will be important to understand the current operations of the inventory control department. Which would be a good technique to understand the workings and issues of that department?

- a. A SWOT analysis
- b. A requirements workshop
- c. Organization modeling
- d. Observation

Q84. John Smith is the business analyst for the new marketing system. This will be his first project for marketing and he is planning an interview with the customer, who is remote. What will be the best way for him to do the interview?

- a. An unstructured interview
- b. An informal conversation
- c. A structured interview
- d. A telephone interview

Q85. The JTC organization is expanding into the online deal market. This is a very competitive market with some strong companies already established. What approach would be good to start understanding the system requirements?

- a. Brainstorming
- b. Focus group
- c. Benchmarking
- d. SWOT analysis

Q86. During an interface analysis effort, the business analyst is looking for a good graphical technique to document the interfaces of a system. Which one of these would work well?

- a. Context diagram
- b. Functional decomposition diagram
- c. Ishikawa diagram
- d. Use case

Q87. When performing document analysis as a part of the elicitation of information for a project, the following would not be a disadvantage:

- a. Using an as-is perspective only
- b. Documentation that may not be up to date
- c. Leveraging previous efforts
- d. Locating information is time consuming

Q88. When doing a formal requirements interview, care should be given to

- a. Only asking previously documented questions
- b. Having an agenda
- c. Starting with open-ended questions
- d. Using process models

Q89. When selecting an elicitation technique, consideration should be given to the geographic location of participants. For a geographically disbursed team, the following would be good to use as one of the techniques:

- a. Brainstorming
- b. Focus group
- c. Observation
- d. Surveys

Q90. When developing a system, one consideration may be to buy a standard off-the-shelf package. Part of the elicitation process should include

- a. Interface analysis
- b. Prioritization of features
- c. A vendor assessment
- d. A requirements workshop

Q91. In a requirements workshop, the team is trying to decide between two alternatives. There are advantages and disadvantages with both, depending on future business developments. The team struggles. What tool would help you organize this?

- a. Decision tree
- b. Traceability matrix
- c. Process model
- d. Solution assessment

Q92. You are planning a requirements workshop. Who would be the best person to lead it?

- a. A business analyst familiar with the business area
- b. A business subject-matter expert
- c. A neutral, experienced facilitator
- d. The project manager

Q93. In order to understand the current situation in the finance department, you will spend 1 week on site learning what goes on in that department and what difficulties the staff are facing. Because they are on the phone a lot of the time, you will need to observe and document but cannot have much interaction with the individuals. What is this technique called?

- a. Passive observation
- b. Active observation

- c. Neutral observation
- d. Remote observation

Q94. You have been assigned as a business analyst for a new initiative and your first task is to interview stakeholders to document their expectations of the initiative. Where should you start?

- a. Designing the interview
- b. Preparing for the interview
- c. Holding an informal short meeting with the interviewees
- d. Getting direction from the sponsor

Q95. You have just started a requirements workshop and you are trying to do a brainstorming session to gather requirements. However, the team is not communicating well and it is difficult to get some interaction going. What should you try first?

- a. Define roles and responsibilities.
- b. Bring in the sponsor.
- c. Be patient; let the team work it out.
- d. Do an icebreaker activity.

Q96. You are to gather requirements for a virtual team, located across the globe. The team has not worked together before. Which of the following deliverables will be most important to create early on?

- a. Risk plan
- b. Communication plan
- c. Quality assurance plan
- d. Change control plan

Q97. You are doing a brainstorming session with a team that has been working together for a while. You know that there are some strong personalities in the group who sometimes dominate the sessions. What would be a good technique to use?

- a. Analytical hierarchy process
- b. Divide into subgroups
- c. Talk to the dominators ahead of the meeting and ask them to hold back
- d. Anonymous brainstorming

Q98. In a workshop for a new order system, requirements will be prioritized. Which of these is a prioritization technique?

- a. Forced-pair ranking
- b. Decision tree
- c. Brainstorming
- d. COCOMO II

Q99. Which of the following is NOT an input into preparing for elicitation?

- a. Stakeholder list
- b. Solution scope
- c. Participants
- d. Business need

Q100. After the elicitation has been performed and documented, the last step in the elicitation process is

- a. Lessons learned
- b. Confirm results
- c. Hand-off to developers
- d. Create prototype

Q101. When doing a passive observation, it is important that the business analyst

- a. Does not let the users know that they are being observed
- b. Explains the purpose of the observation
- c. Discusses the level of knowledge and performance of the observed users with their manager
- d. Documents neutral observations

Q102. When using prototyping as a technique to elicit requirements, special care must be given to

- a. Technical accuracy
- b. The availability of developers
- c. Feasibility of the solution
- d. Realistic stakeholder expectations

Q103. When studying industry best practices, this is often an issue

- a. Availability of information
- b. Resistance from user to looking at the outside
- c. It takes too long
- d. It may be illegal

Q104. A JAD session can be full of distractions and loss of focus. Which of these will best help to control conflicts?

- a. An agenda
- b. A clear goal
- c. Ground rules
- d. A speech by the sponsor

Q105. Which guideline must be kept in mind when designing a requirements survey?

- a. Keep it simple
- b. Allow for complete answers to avoid misunderstandings
- c. Do follow-up on unclear answers
- d. Encourage participation

Q106. One good way to prepare for requirements gathering is to study the existing systems. One good advantage of this is

- a. It can be good groundwork if the user has limited availability.
- b. It shows how the business is done.
- c. It takes little time.
- d. It gives good ideas for improvement needs.

Q107. Which one is the biggest driver when selecting an elicitation technique?

- a. Team location
- b. Customer
- c. Skill set
- d. Development approach

Q108. A big advantage of brainstorming is

- a. It is objective.
- b. It is easy to stay focused.
- c. It generates a lot of ideas in a short time.
- d. It is good for analysis.

Q109. When preparing for an interview, it is best if both the interviewer and interviewee are clear on

- a. Interview styles
- b. The development process
- c. The as-is situation
- d. The boundaries of the interview

Q110. Ravi is preparing for a requirements workshop to create a logical data model.

This is the users' first such session. It will be important that Ravi

- a. Establish ground rules.
- b. Have clear objectives.
- c. Does data modeling training with the participants as needed.
- d. Create a draft before the session.

Q111. Which of these projects would make a good candidate for a requirements workshop?

- a. Multi-stakeholder project with disagreements expected
- b. Short, simple efforts
- c. Projects that lack stakeholder commitment
- d. Projects where the scope is well defined

Q112. Density dotting is a popular group technique. What does it do?

- a. Finds bottlenecks
- b. Analyzes processes
- c. Prioritizes
- d. Tracks user population

Q113. A sponsor may play an important role in the requirements workshop. What would be one of the main benefits of bringing the sponsor into the session?

- a. To audit team performance
- b. To provide management's view on the requirements
- c. To stop conflict
- d. To show the importance of the project

Q114. What is the main difference between a focus group and a requirements workshop?

- a. A focus group addresses only one topic.
- b. A focus group is not a decision-making group.
- c. A focus group has only people with the same background.
- d. A focus group is normally smaller.

Q115. The workshop is going well, the team has been working for a couple of days, and the discussion is flowing. What should the facilitator do?

- a. Let the team work; it is now in the performing stage.
- b. Play devil's advocate to stir things up a bit.
- c. Take a break and check status with the individuals.
- d. Change the team a bit to add new perspectives.

Q116. Which of the following is a tool to help keep a session focused?

- a. Data model
- b. Parking lot
- c. Icebreaker
- d. Lessons learned

Q117. The project team has decided to do a series of virtual requirements workshops using a web meeting for participants from three countries. Which of the following would increase the chance of success the most?

- a. Have a goal for each session.
- b. Have a face-to-face meeting before the workshop.
- c. Train people on the process.
- d. Invite the sponsor to the kickoff.

Q118. Which of the following would NOT be a good reason to do a requirements prototype?

- a. To evaluate technical feasibility
- b. As a proof of concept
- c. For communication
- d. To reduce abstractness

Q119. When selecting storyboarding as a technique to capture requirements, what would be a disadvantage?

- a. The customer does not relate.
- b. It can only be used late in the process.

- c. It has a different look and feel than the final product.
- d. It is mainly for developers.

Q120. When using an iterative approach to requirements gathering, it is important that

- a. The iteration covers the whole life cycle.
- b. There is only one iteration per requirements type.
- c. Prototypes are developed for each iteration.
- d. Each iteration has its own goal.

Q121. Observers may be invited to a requirements workshop to

- a. Assess the performance of the participants.
- b. Contribute ideas.
- c. Learn for future projects.
- d. You should never have observers in a workshop.

Q122. The selection of elicitation technique will be driven by

- a. Stakeholder availability and preference
- b. Development approach
- c. Business analyst experience
- d. Organizational standards

Q123. Job shadowing can help document

- a. Requirements priorities
- b. SWOT analysis
- c. Data models
- d. Process flow

Q124. When eliciting requirements, it must be remembered that requirements elicitation is normally

- a. Frustrating
- b. Iterative
- c. Done in a group setting
- d. Focused on the end user

Q125. Observation must always be combined with other elicitation techniques because

- a. Observing everything takes too long.
- b. Observation is an as-is technique.
- c. Observation only gives one person's view.
- d. It keeps the process fair and balanced.

Q126. After the first 30 minutes of the workshop, there is starting to be some conflict between participants. What should the facilitator do?

- a. Nothing. It is normal in this setting.
- b. Stop the session and hold one-on-one interviews with the parties.
- c. Intervene and do a conflict analysis.
- d. Let the team work it out as long as ground rules are followed.

Q127. For an upcoming project, there are only two stakeholders. They are at a high level in the organization and located in different regions. Which elicitation technique is probably best?

- a. Observation
- b. Requirements workshop
- c. Interviews
- d. Document review

Q128. When documenting the results of a workshop, documentation should include

- a. Decisions and rationale for decisions
- b. All models
- c. Pictures of whiteboards and flipcharts
- d. Everything discussed in the session

Q129. Tracking issues is key for a requirements workshop. Unless otherwise stated, this should be assigned to

- a. The facilitator
- b. The business analyst
- c. The project manager
- d. The sponsor

Q130. What is a focused event with a few stakeholders for a short, intense period?

- a. Focus group
- b. Walkthrough
- c. Requirements workshop
- d. Gate review

Q131. As the requirements for the new Swede-Mart project are being presented, there is strong disagreement between the stakeholders. The business analyst is getting frustrated because the stakeholders have had access to the document for 2 weeks, but have not raised any issues previously. Which of these statements is most accurate?

- a. This is a project manager issue, because the resources have not done what was expected of them.
- b. The business analyst should stop the meeting and ask everyone to read through the document and provide feedback before the next meeting.

- c. The problem should be escalated to the functional manager to make sure that the job gets done for the next meeting.
- d. The business analyst should review the requirements using a process that ensures that the stakeholders understand the requirements.

Q132. What is the right level of detail to use when communicating requirements?

- a. The level of detail must be low enough to avoid any misunderstandings.
- b. The level of detail should be based on the needs of the stakeholder.
- c. The organization should use a standard for what level of detail is needed so that all projects are consistent.
- d. The level of detail should be what the developers need to develop specifications.

Q133. When creating reusable requirements, care should be given to

- a. Ensuring that the requirement is fully tested before it is documented in a repository
- b. Making them detailed enough to be used by the developers
- c. Expressing them in terms that make them usable in the long term
- d. Having them approved by all stakeholders

Q134. To trace requirements for a project we may use a

- a. Coverage matrix
- b. CRUD matrix
- c. Process model
- d. Prototype

Q135. On the new sales support project, Hiram is trying to organize the requirements and prioritize them. In order to do that, Hiram should first

- a. Determine the relationships between the requirements.
- b. Ask the customer to give a prioritized must have/want to have list.
- c. Get an estimate for the cost of each requirement.
- d. Do a draft prioritization to bring to the customer.

Q136. Once a requirements document has been baselined, then any changes to the requirements

- a. Are not allowed
- b. Must be requested by the customer and approved by the developers
- c. Must go through formal change control
- d. Must be communicated to all stakeholders

Q137. During the later portion of the requirements elicitation stage of the order entry upgrade project, disagreement about requirements causes conflict

between the purchasing and receiving departments. What would be an appropriate action by the business analyst?

- a. Since purchasing has the most at stake on this project, their voice is more important.
- b. Escalate to the sponsor.
- c. This is normal and will get solved during the design phase.
- d. This should be handled by the project manager, not the business analyst.

Q138. When the business analyst is preparing a requirements package to send out to potential vendors and when the organization is open to new ideas, then it would be best to issue an

- a. RFP
- b. RFQ
- c. RFI
- d. RFS

Q139. When creating a requirements package for the development team, a main consideration should be

- a. How well does the requirements package support the solution development?
- b. How technically feasible is the requirements package to implement?
- c. How well does the requirements package solve the business need?
- d. How well does this requirement package integrate with other systems?

Q140. What would be a difficulty in attempting requirements reuse?

- a. Less flexibility in implementing requirements
- b. Difficult to find and retrieve unless the process is formalized
- c. Different business units may want to use different functionalities
- d. Developers resist standard solutions

Q141. When preparing for sign-off of the requirements document, consideration must be given to

- a. Always getting approval in writing
- b. Making sure the person paying for the project is the one signing
- c. Mandatory signing only if regulation demands it
- d. Who has authority to sign what portion of the document

Q142. What is the definition of a *requirements deliverable*?

- a. Any deliverable in the requirements phase
- b. Anything included in the business requirements document
- c. A deliverable used as a basis for solution development and implementation
- d. A prototype or other product used to elicit requirements

Q143. John has been asked to organize requirements and define the attributes necessity and effort. What would this help with?

- a. Traceability
- b. Buy-in
- c. Prioritization
- d. Design

Q144. When identifying issues with requirements during the approval process, this technique would be useful:

- a. Tracing
- b. Problem tracking
- c. Process modeling
- d. Risk analysis

Q145. When using a change-driven project approach, the following is normally not implemented:

- a. Status meetings
- b. Problem tracking
- c. Formal change control
- d. Audits

Q146. What is the difference between solution scope and project scope?

- a. Solution scope is a subset of project scope.
- b. Solution scope will drive large portions of the project scope.
- c. Solution scope and project scope are independent.
- d. Project scope will determine realistic solution scope.

Q147. When looking at the need for detail in requirements, the end users will need

- a. Easy-to-understand requirements because they will be directly impacted
- b. To understand how the technical solution will be implemented
- c. Only a high-level understanding since they will receive training on the actual product
- d. To be the experts on the product

Q148. A formal presentation may be best when presenting requirements to

- a. Check status
- b. Communicate requirements to business areas
- c. Communicate requirements to other project teams
- d. Make decisions regarding scope

Q149. This stakeholder should normally receive summary level requirements communication

- a. Project manager
- b. Sponsor

- c. End user
- d. Implementation SME

Q150. When should approval of requirements take place?

- a. At the end of the requirements phase
- b. As the requirements document is handed over to the design team
- c. At different stages depending on the stakeholders
- d. In the project charter

Q151. Allocation of requirements is a part of

- a. Business analysis planning
- b. Requirements elicitation
- c. Requirements analysis
- d. Forward traceability

Q152. After requirements are organized and analyzed, the relationships between requirements should be documented. One of the relationships defined could be *necessity*. What would that most likely be used for?

- a. Grouping requirements for implementation
- b. Eliminating requirements
- c. Traceability
- d. Development of business case

Q153. Who would be a main stakeholder in maintaining requirements for reuse?

- a. End user
- b. Testers
- c. Customer
- d. Project manager

Q154. When in the project cycle is a requirements package prepared?

- a. At the end of the requirements phase
- b. When the customer has signed off on the requirements
- c. Throughout the process
- d. When communication is needed

Q155. What must a requirements package to the testers include?

- a. Functional and nonfunctional requirements
- b. Test cases
- c. Stakeholder requirements
- d. Expected results

Q156. When developing an RFP, what would be a good guideline?

- a. Make sure that the vendor receives specific exact questions.
- b. Only evaluate responses based on technical and business fit.

- c. Avoid closed-ended questions.
- d. Use a wide selection of vendors.

Q157. Which is true about requirements communication?

- a. All requirements communication should be planned.
- b. It should be done iteratively.
- c. It should be kept separate from elicitation.
- d. It should normally be done as a separate activity.

Q158. When communicating interface requirements between different departments, this diagram often works well:

- a. Data flow diagram
- b. Activity diagram
- c. Sequence diagram
- d. Use case diagram

Q159. Which of the following would NOT be a part of a requirements review package?

- a. Models
- b. Text
- c. Requirements approach
- d. Prototype

Q160. What is a work product for a business analyst?

- a. The same as a deliverable.
- b. Notes and diagrams used during the requirements development process.
- c. Something concrete such as a prototype.
- d. Results from a business analysis process.

Q161. Which one of these diagrams is primarily focused on establishing the scope of a product?

- a. Context diagram
- b. Functional decomposition diagram
- c. Entity relationship diagram
- d. Data flow diagram

Q162. Your manager has asked you to identify potential business needs and to present a plan for how to do it. Which one of the following techniques would not be helpful in identifying new business needs?

- a. Assess current state of technology.
- b. Review project charters.
- c. Do a benchmark analysis.
- d. Engage in competitive studies.

Q163. Why do you need to perform an strategy analysis before starting a new project?

- a. It is a mandatory deliverable by the IIBA.
- b. It defines from whom we need to solicit requirements.
- c. It ensures buy-in from the project sponsor.
- d. It defines an appropriate scope for the solution.

Q164. You are working with senior management to evaluate a potential opportunity. You have been asked to facilitate a session where the goal is to get an idea of the main facts impacting the decision. Which technique would work best?

- a. SWOT analysis
- b. Opportunity analysis
- c. Decision analysis
- d. Problem analysis

Q165. During a problem analysis session, the discussion turns to new products in the market place that may help solve the problem being discussed. The products seem to be wide ranging and deal with very different issues. When you bring that up, you get some initial blank stares and one of the participant's says, "Well, right now we don't know what the real issue is. There are many things that cause this problem, and all of these products would deal with different ones of those issues." You suddenly realize that the session has skipped from problem identification to looking for a solution. What critical step have you missed?

- a. Stakeholder buy-in
- b. Root-cause analysis
- c. SWOT analysis
- d. Problem definition

Q166. Once the solution scope has been defined, then the next step would often be to create which document?

- a. Business need definition
- b. Problem analysis summary
- c. Business case
- d. Enterprise architecture

Q167. What is the business analyst's role in defining the solution scope?

- a. He may help but this is really a project management task.
- b. He is responsible for describing the business environment and the business requirements.
- c. Until the project manager is assigned, he will function as the interim project manager.
- d. No involvement. The business analyst will get involved after the scope is defined.

Q168. You are doing a root-cause analysis with the customer for a problem that they have experienced for the last 2 years. This customer is very structured and organized but tends to lose patience easily. What technique would you use?

- a. Brainstorming
- b. A simple problem analysis template
- c. Five why's
- d. Fishbone diagram

Q169. You have been called into a meeting with the new CEO of your company. He has tasked you with identifying the current state of the business. The company has not done well recently and he is looking to set a new vision and direction for the company. He wants to make sure that he has a solid understanding of the current business architecture. The focus is not on time or cost, but to get it right. You go back to your office to come up with a recommended approach. Which of these options would you recommend?

- a. Create a root-cause analysis model to understand the root cause of the current business problems.
- b. Organize a Zachman framework model to understand the business as it is today.
- c. Perform benchmarking against the main competitors to evaluate what they are doing better than your company.
- d. Do a gap analysis to determine the gaps between where the company is and where it should be.

Q170. As a part of the strategy analysis effort going on within your business unit, your manager (who just attended an IIBA-sponsored conference) asks you to develop business use cases for the organization. When you push for more details, it is clear that the manager heard about them at the conference, but has no real understanding of what they are. You do some research and find

- a. This is really a great idea. Business use cases should always be a part of describing the as-is situation in an organization.
- b. It may be great to create a business use case, but this is not the business analyst's job. Business use cases should be created by the business and then the business analyst will derive a system use case from them.
- c. This could be really good. Business use cases help describe the key functions and services that the business provides for its stakeholders.
- d. Once again, it proves your view that sending managers to conferences is a waste of time and money. Business use cases will be used in requirements analysis, not in strategy analysis.

Q171. The project you are working on has determined the business need. You are in a meeting with the stakeholders and are trying to determine the next step. What would be your recommendation?

- a. Determine the gap that exists in the organization as far as dealing with this business need is concerned.
- b. Determine the recommended solution.
- c. Develop the business case.
- d. Confirm strategic goals.

Q172. When developing a business case, which one of these deliverables would typically not be included?

- a. Risk assessment
- b. Cost estimate
- c. Benefit estimate
- d. Solution scope

Q173. What role would the data model play in strategy analysis?

- a. A conceptual data model would be very useful in understanding the business.
- b. A detailed as-is data model would be needed to do a good gap analysis.
- c. Wait until the requirements analysis phase to use the data model.
- d. A data model can show dynamic changes to the business.

Q174. Which of the following would be included in the business architecture?

- a. Software applications
- b. Hardware
- c. Business goals
- d. Decision support tools

Q175. You have been asked to perform a feasibility study. What is that?

- a. Studying the technical feasibility of a new solution
- b. Evaluating the feasibility of a vendor solution
- c. Another word for business case
- d. Preliminary study of solution options and business benefits

Q176. When documenting a problem, the following should be included:

- a. Description and who caused it
- b. Description, who is affected, and who is impacted
- c. Description of the problem and a proposed solution
- d. Where, when, and how the problem occurred

Q177. Which of the following is not an input into defining the business case?

- a. Business need
- b. Solution scope

- c. Stakeholder concerns
 - d. Functional requirements
- Q178. After successfully completing a number of business analysis projects, you have been asked to recommend a solution approach for the company's latest initiative. An example of that may be
- a. Change the business procedure
 - b. An agile development approach
 - c. Object-oriented modeling
 - d. Outsource the development
- Q179. When defining the solution scope care must be given to identify
- a. Impacted organizations
 - b. Financial benefits
 - c. End user
 - d. Technical platform
- Q180. Gap analysis means
- a. The gap between the technical solution and customer requirements
 - b. The gap between customer expectations and the product being developed
 - c. The gap between the current situation and the to-be situation
 - d. The gap between vendor product capability and customer requirements
- Q181. A business case should include
- a. Quantitative evaluations of the benefits and cost of a project
 - b. Qualitative evaluation of the benefits and costs of a project
 - c. Qualitative and quantitative evaluations of the benefits of a project
 - d. Vendor estimates and assessments
- Q182. An initial business case should include a risk assessment. Which one of these elements would normally NOT be a part of that?
- a. Financial risk
 - b. Organizational risk
 - c. Project manager risks
 - d. Feasibility risks
- Q183. A good example of a business event is
- a. A product recall
 - b. Creating data backup
 - c. Sending in taxes
 - d. Updating inventory
- Q184. Business rules are used to describe how the business operates. Which statement is true about the relationship between requirements and business rules?
- a. A requirement may change a business rule.
 - b. A business rule is always a requirement.

- c. All requirements trace to business rules.
- d. A requirement cannot contradict a business rule.

Q185. In order to perform which of the following will you need to analyze different solution options and see how they address the business problem at hand?

- a. A business case
- b. A feasibility study
- c. Business needs analysis
- d. Enterprise architecture

Q186. Which of the following conceptualizes the recommended solution in enough detail that the new business capabilities of an effort will be understood?

- a. Solution scope
- b. Business need
- c. Context diagram
- d. Problem statement

Q187. To understand the information capabilities of an enterprise, we must understand its networks and communication links. Which model would work to document this?

- a. Process model
- b. Infrastructure model
- c. Technology model
- d. Information model

Q188. A business entity model can be useful

- a. For modeling interactions between businesses
- b. For enterprise data models
- c. To model the information needs of an organization
- d. To model core business data

Q189. Which one of these activities is not part of strategy analysis?

- a. Determining the most feasible solution option
- b. Assessing the capabilities of the organization
- c. Tracing the business need to the solution
- d. Documenting business requirements

Q190. Why do we define the business need?

- a. To trace the solution back to it
- b. To identify and define why we need to change
- c. To understand the project scope
- d. To see if the project is a good idea

- Q191. One example of an implementation approach is
- Implementing in multiple releases
 - Using case modeling
 - The change-driven approach
 - Workshop focused
- Q192. To define solution scope, Jim has decided to identify the key stakeholders and model their goals for the solution. Which techniques would allow this?
- Process modeling
 - Enterprise modeling
 - Event modeling
 - User stories
- Q193. Which of the following techniques works well when identifying solution approaches for mature business areas?
- Process modeling
 - Enterprise modeling
 - Prototyping
 - Benchmarking
- Q194. Which of these models is an example of documenting enterprise architecture?
- Infrastructure model
 - Zachman framework
 - SWOT analysis
 - Goal model
- Q195. When documenting impact in a problem statement, it is important to
- Quantify it.
 - Look at the impact of the solution.
 - Look at the impact on each stakeholder.
 - Review the impact on the business as a whole.
- Q196. When doing a SWOT analysis one should consider
- Capitalizing on strengths
 - Hiding weaknesses
 - Looking at the impact on each stakeholder
 - Looking at it with an outsider's view
- Q197. The types of capabilities that are most likely to be evaluated during an assessment of the capability gap include
- Development skills
 - Available packaged solutions
 - Staff competencies
 - Financial capabilities

Q198. An Ishikawa diagram may be used to assist in what activity?

- a. Assessing capability gaps
- b. Defining solution scope
- c. Defining the business case
- d. Defining the business need

Q199. The business case includes an initial risk assessment. What is the purpose of this risk assessment?

- a. To evaluate if the initiative is riskier than the organization is willing to tolerate
- b. To determine the uncertainties in benefits and cost
- c. To identify market risks for the product
- d. To determine the level of contingency needed for the initiative

Q200. To evaluate the viability of an initiative, the organization may use net present value. What is the basic idea behind net present value?

- a. All revenues minus all costs
- b. The value of the current solution versus the value of the future solution
- c. Comparing the value of a vendor solution versus custom development
- d. Comparing all revenues and costs based on today's value of money

Q201. Project A has a benefit of \$100,000 and a cost of \$50,000. Which statement would be true based on this?

- a. The NPV is \$50,000.
- b. The BCR is 2.
- c. The payback period is 6 months.
- d. The IRR is 10 percent.

Q202. Which of the following would be a constraint?

- a. The solution will be able to run on existing infrastructure.
- b. Any solution must use existing infrastructure.
- c. The current infrastructure will be sufficient to handle the workload from the new solution.
- d. The solution will improve processing capability by 10 percent within 1 year of implementation.

Q203. Which of these estimating techniques is most likely to be used during business case development?

- a. Bottom-up
- b. Rolling wave
- c. Analogous
- d. Function point counting

Q204. The solution scope must be sufficiently defined during strategy analysis to be able to

- Develop a business case.
- Select a solution.
- Develop a project charter.
- Determine the need for organizational change.

Q205. Payback period can be defined as

- The time it takes to get the investment back
- How long after implementation the project starts to make a profit
- How fast the vendor expects payment for its services
- How soon the project objectives have been met

Q206. There have been issues with the processing times for orders at JTC Enterprises. To find out more about the symptoms of the problems and to investigate causes, the business analyst could identify and meet with

- The support organization
- The end user
- The sponsor
- A focus group

Q207. For what can you use the analytical hierarchy process?

- Flowcharting
- Requirements modeling
- Decision making
- Root-cause analysis

Q208. Business rules should not

- Be documented for a specific project
- Include operative rules
- Include information about how they are enforced
- Relate to data needs

Q209. Which statement is true about cash flow?

- Cash flow is not something that would concern the business analyst.
- Although the business analyst provides input into cash flow analysis, it is not his or her primary concern.
- Cash flow is only handled by the finance organization.
- Cash flow should always be a part of the business case.

Q210. Which elicitation technique would be good to start with for determining the business need?

- Document analysis
- Requirements workshop

- c. Focus group
- d. Brainstorming

Q211. Which one of the following would typically not be a primary actor in a system use case?

- a. System user
- b. A different system
- c. The shareholders
- d. Time

Q212. When reviewing the level 2 data flow diagrams (DFDs) that were created by the business analyst, the customer noticed that there was data coming in from the federal government, which had not been previously specified. How would you respond to that if you were the business analyst?

- a. Good catch. We will submit it as a change control to the level 0 and level 1 DFD and evaluate the impact.
- b. Good catch. We will update the level 0 and level 1 DFD.
- c. It is normal that you discover more data flows as you decompose a DFD. It is just part of the process.
- d. As long as a data flow from an external entity, over which you have no control, shows up in at least one level of the DFD, you are OK.

Use [Figure 11.1](#) to answer questions 213 through 215.

Q213. Review [Figure 11.1](#). Which of the following statements is true about the relationship between the order and the customer?

- a. The customer must place at least one order.
- b. The order may be placed by one customer.
- c. The customer must place many orders.
- d. The order can only be placed by one customer.

Q214. Review [Figure 11.1](#). Which statement is true about the customer and order items?

- a. The customer places zero or many order items.
- b. There is no relationship between the customer and the order item.
- c. The customer places one or many order items.
- d. The customer places one and only one order item.

Q215. During a JAD session, the customer mentions that orders are often entered in the system before they have figured out who the customer is. Eventually, the customer will be added but it could be quite a while before that happens. How would you change the model in [Figure 11.1](#) to reflect this?

- a. It does not need to be shown in the model. The customer can just add a pending customer number that will be changed later.
- b. Change the minimum cardinality from the customer to the order to be mandatory.

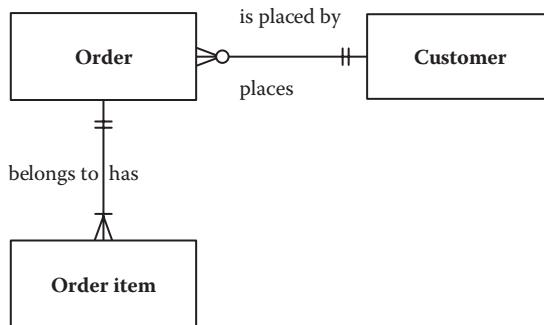


Figure 11.1 Entity relationship diagram.

- c. This is a processing issue that will be reflected in the process model. The ERD only shows static data needed.
 - d. Change the cardinality from the order to the customer to be zero or one.
- Q216. A good model is normally made up of two components. What are they?
- a. The notation technique and the business it is describing
 - b. A user view and a technical view
 - c. A diagram and a text document
 - d. The notation technique and the tool
- Q217. The system you worked on for the last 2 years has just been successfully implemented. The customer agrees that it meets all the requirements that had been stated. However, the business problems that led to the start of the project are still evident. Which process should have dealt with this?
- a. Determining assumptions and constraints
 - b. Root-cause analysis
 - c. Validating requirements
 - d. Verifying requirements
- Q218. A key concept in working with the customer to capture, analyze, and document requirements is to use progressive elaboration. This term, borrowed from project management, is intended to send what message?
- a. That the customers normally do not know what they want so the business analyst needs to elaborate.
 - b. That an iterative approach is often appropriate, with each iteration elaborating the requirements further.
 - c. That the requirements must first be documented as *whats* before you document the *hows*.
 - d. That the business analyst must work with the solution experts to elaborate appropriate technical solutions to business problems.

Q219. Your project manager, just returning from a conference, has decided to discuss data modeling with you. She is asking if you are creating a conceptual or a logical data model. What is the difference?

- a. Just different terminology. One is using information engineering; the other is using the object-oriented approach.
- b. The conceptual data model is used in the analysis phase, and the logical data model is used in the design phase.
- c. The conceptual data model is high level, not showing attributes, sub-types, or cardinality. The logical data model is a fully defined and normalized to third normal form.
- d. The conceptual data model is used to communicate with the customer, and the logical data model is used to communicate with the developer.

Q220. For an upcoming JAD session with the customer, the developers request that you develop use cases since they will be doing development using a tool based on use cases and object-oriented development. You tell them that you need some time to consider their request. What should you answer?

- a. No way. You will pick the techniques best suited for the customer.
- b. You agree. The organization's investment in the tool requires a commitment from all areas of the business.
- c. You recommend that they use use cases for the design. It is not really an analysis technique.
- d. You agree to use the diagram but not the descriptions. That would be too detailed for the customer.

Q221. Which one of the following statements is not considered a best practice when writing textual requirements?

- a. Writing independent requirements that are not connected to other requirements.
- b. Writing only one requirement per statement.
- c. Describing who is fulfilling the requirement.
- d. Not assuming that the reader understands the business.

Q222. Toward the end of the analysis phase of the project on which you have been working, you realize that there is a need to convert data from the old system to the new system. What type of requirement is this?

- a. Functional requirement
- b. User requirement
- c. Implementation requirement
- d. Optional requirement

Q223. You have been asked to review the requirements, analyze them, and trace them back to business objectives. Which requirements analysis task are you performing?

- a. Prioritizing requirements
- b. Modeling requirements
- c. Verifying requirements
- d. Validating requirements

Q224. In a peer review of the business requirements document, the subject-matter expert brings up that, based on the processing needs of different locations and the sheer volume of data, they do not think the current, normalized, data model is the right solution. It needs to be less divided in order for the customer to write good queries. How do you deal with this?

- a. You do nothing. It is not your problem.
- b. You start by normalizing the data, then de-normalize on the basis of the constraints identified.
- c. You document the de-normalized database as a constraint in the business requirements document.
- d. You make a note in the business requirements document but do nothing.

Q225. You are meeting with the customers to discuss their business problem. They mention that these problems are mainly happening at certain times of the year when the staff is under a lot of pressure from stressful activities. What type of modeling may help you understand their requirements?

- a. Data modeling
- b. Event modeling
- c. Use case modeling
- d. Organizational modeling

Q226. While creating a use case model with the customer, you realize that some of the functionality within the use case will not be implemented in the first release of the product but at some future, not yet determined, date. How would you deal with that?

- a. Put them in an includes use case to be included later.
- b. You should not model them. Only the current scope should be included.
- c. Put them in an *extends* use case to extend the functionality later.
- d. Include them in the regular model. The model is intended to show solution scope, not project scope.

Q227. In your first requirements-gathering session with the customers, it is clear that they do not fully understand what business processes are affected by the project you are working on. Their discussion appears to be at a very high

level, and you are trying to get them down to a lower level of detail. Which modeling technique may help you with this?

- a. Class model
- b. Use case model
- c. Data model
- d. Functional decomposition diagram

Q228. You have been asked to create a CRUD matrix for the requirements modeling effort on your project. What does this show?

- a. Process responsibilities
- b. Physical data
- c. Ownership of data
- d. Use case ownership

Q229. Written requirements should be verifiable. This means

- a. A test plan should be done.
- b. We can verify that the requirement is met after the solution is done.
- c. They can be verified with the users for accuracy.
- d. They can be traced back to a business need.

Q230. In an order entry system, the order can be open, in process, shipped, paid, or closed. Which model would help document when an order moves from one type to the other to another?

- a. State diagram
- b. Class diagram
- c. Sequence diagram
- d. Activity diagram

Q231. Lucy is documenting requirements using use cases. She has collected a number of requirements that are performance related, such as system availability, number of users, and expected response time. Where should she document these?

- a. In the use case body.
- b. In the process model.
- c. In a nonfunctional requirements document.
- d. These should be captured by the design team.

Q232. Lisa is gathering requirements for a new Android gaming application. She wants to capture different user scenarios that will be fast to develop and easy to walk through with sponsor and developer. Which technique may work well?

- a. Storyboards
- b. Use cases
- c. Data flow diagrams
- d. Prototypes

Q233. In a document from the customer regarding developing the new mortgage application, you learn that “Any customer who is late with a payment of more than 15 days will be assessed a penalty.” What would be the best classification of this information?

- a. Constraint
- b. Assumption
- c. Requirement
- d. Business rule

Q234. Which of the following will work well to show how information is passed between objects in a system?

- a. Use case diagram
- b. Object diagram
- c. State diagram
- d. Sequence diagram

Q235. While class diagrams and entity relationship diagrams (ERDs) look similar, they also have some key differences. What would be one example of this?

- a. A class diagram does not show cardinality; ERD does.
- b. A class diagram shows physical storage; ERD does not.
- c. A class diagram shows operations on the data; ERD does not.
- d. A class diagram does not work well with use cases; ERD does.

Q236. When documenting a postcondition for a use case, what should this show?

- a. What state the system is in at the end of the use case
- b. What process is following the use case
- c. How the end user knows that the use case has ended
- d. What condition ends the use case

Q237. A traditional way of showing business functions and processes from an internal view is

- a. Object diagram
- b. Functional decomposition diagram
- c. Workflow diagram
- d. Use cases

Q238. When creating a swim lane diagram, this is a rule to be followed:

- a. Only one end point
- b. Only one start point
- c. Only people can have swim lanes
- d. Loops are not allowed

Q239. UML models are divided into which of the following three main groups?

- a. Data, process, flow
- b. Internal, external, transitional

- c. Structure, behavior, interaction
- d. Use case, class model, activity diagram

Q240. A primary reason to do modeling on a project is

- a. Communication
- b. To automate software generation
- c. To create traceability
- d. Correct documentation

Q241. When working on a new system development project, you are asked to create a conceptual data model. What does this mean?

- a. A model that shows business entities only
- b. A model that shows relationships but not cardinality
- c. A model that is used by the system architects
- d. A model that is used for communication with the customer

Q242. When creating a data flow diagram, what is true about the data store?

- a. It collects data for some time period.
- b. It is the same as an entity in the data model.
- c. It will become a physical database.
- d. It describes a software view of the data.

Q243. An entity relationship diagram is made up of

- a. Entities, relationships, cardinality
- b. Entities, operations, attributes
- c. Entities, attributes, relationships
- d. Entities, attributes, methods

Q244. A relationship in an ERD can be viewed as

- a. A clarification
- b. A decomposition
- c. An association
- d. Two business rules

Q245. Use cases are different from traditional modeling in that they

- a. Do not look at data, just process.
- b. Take an external view of the system.
- c. Do not use a pictorial view.
- d. Are only used for the software portion of the solution.

Q246. Entity life cycle diagram is another name for

- a. State diagram
- b. Entity relationship diagram
- c. Sequence diagram
- d. Data flow diagram

Q247. In a customer meeting gathering requirements for the new online store your company is setting up, the customer states that “once the customer enters their billing information, we should initiate a credit check. However, the customer should be able to continue his or her processing while the credit check is being processed.” How could you best model this?

- a. By starting another activity diagram for the credit check process
- b. As an asynchronous flow in a sequence diagram
- c. As an alternative path in a use case
- d. As an event in a state diagram

Q248. What does a user story include at a minimum?

- a. Description, exceptions, error handling
- b. Description, happy path, alternative paths
- c. Description, benefits
- d. Description, screen layout

Q249. The product must run on Windows 2007. Where would you best document this customer request?

- a. As a nonfunctional requirement
- b. As a regulatory requirement
- c. As an assumption
- d. As a constraint

Q250. When creating a functional decomposition diagram, which advice would be best?

- a. Stay at a high level; too much detail will get confusing.
- b. When working with executives, use a top-down approach; with end users, create a bottom-up view.
- c. At the lowest level, clarify the relationships between the processes.
- d. Do a top-down or bottom-up approach based on whether you are documenting as-is or to-be conditions.

Q251. A data model has been defined with a vehicle entity called *Vehicle*. Two subtypes are defined: *Car* and *Truck*. Which statement is true?

- a. *Car* and *Truck* will inherit all attributes and relationships from *Vehicle*.
- b. *Car* and *Truck* must have their own attributes.
- c. *Car* and *Truck* will inherit all mandatory attributes and relationships, but not the optional ones.
- d. *Vehicle* will be a sum of all the attributes from *Car* and *Truck*.

Q252. When verifying requirements, one of the criteria to look for is testability. What does this mean?

- a. That a test case accompanies the requirement
- b. That a test case has been designed

- c. That a test case can be designed
- d. That we know how to test it in the solution

Q253. The process of validating requirements means that the stakeholders, solution, and transition requirements align with

- a. Strategic goals
- b. Business requirements
- c. The charter
- d. The product

Q254. A secondary actor in a use case is

- a. Someone who receives output from the use case
- b. Someone providing input to the use case
- c. A system that assists with processing for the use case
- d. All of the above

Q255. What is a temporal event?

- a. An event that will go away after awhile
- b. An event triggered by the passage of time
- c. An event that must occur on a certain date
- d. An event that is triggered by temperature

Q256. When decomposing a data flow diagram you must

- a. Keep equality at the borders.
- b. Add data stores at the lower levels.
- c. Connect it with the functional decomposition diagram.
- d. You cannot decompose a data flow diagram.

Q257. When performing interface analysis, the primary focus is on

- a. User interfaces
- b. Interfaces to external applications
- c. Interfaces to external hardware
- d. All types of interfaces

Q258. All processes in a data flow diagram must have

- a. Connection to other processes
- b. Inputs and outputs
- c. Connections to data stores
- d. Connections to external agents

Q259. When selecting an easy to understand intuitive modeling technique that works well with customers, consider using

- a. Sequence diagram
- b. Class diagram
- c. Data flow diagram
- d. Entity relationship diagram

Q260. You are working on documenting data requirements. One stakeholder says, “A customer may place an order.” Another wants the statement “A customer places an order.” Does it matter?

- a. Yes; the first statement would allow customers who have never placed an order to do so; the second would not.
- b. There is no difference; the relationship is the same.
- c. You should dive deeper to see if there really is a difference.
- d. Yes; the second case is right. You cannot have a customer who has not placed an order.

Q261. What is a prototype called, which is developed with the intent that it becomes the final product?

- a. Progressive prototype
- b. Evolutionary prototype
- c. Working prototype
- d. Throw-away prototype

Q262. What causes an object to change its state in a state diagram?

- a. A trigger
- b. An action
- c. An event
- d. An intervention

Q263. One disadvantage with a sequence diagram is

- a. User processing is often not sequential.
- b. It normally requires a fully defined class model.
- c. It does not work with use cases.
- d. It does not show how objects interact.

Q264. What is a scenario in a use case?

- a. It sets the stage with precondition, postcondition, and actions.
- b. Possible error conditions.
- c. A graphical illustration of flow.
- d. One way for the actor to meet the goal.

Q265. Which of the following models shows how people or groups collaborate over time to perform work?

- a. Process model
- b. Event model
- c. Business model
- d. Data flow diagram

Q266. Which of the following would NOT be a nonfunctional requirement?

- a. Security
- b. Reliability

- c. Updating information
- d. Maintainability

Q267. What is the best description of a model?

- a. A graphical representation of a complex situation
- b. A text representation of a piece of a system
- c. A simplified representation of a complex reality
- d. A description of a problem

Q268. Which of the following would NOT be a recommendation when documenting text requirements?

- a. Express only one requirement at a time.
- b. Assume that the reader has basic domain knowledge.
- c. Write in the active voice.
- d. Describe a problem.

Q269. While working on interface analysis, you have identified a number of interfaces to systems owned by other functions in your organization. This is an indication that you will need to use this fundamental skill:

- a. Negotiation
- b. Facilitation
- c. Persuasion
- d. Interview

Q270. In a data model, two related entities will have

- a. Two and only two relationships
- b. One and only one relationship
- c. At least two relationships
- d. At least one relationship

Q271. While evaluating the need to replace an existing system, the business analyst should make sure to evaluate

- a. Sunk cost
- b. Opportunity cost
- c. New technology
- d. What the end user wants

Q272. When doing solution validation, we should make sure to verify that

- a. The solution meets the business need.
- b. The solution meets requirements.
- c. The customer is satisfied with the solution.
- d. All contractual issues are solved.

Q273. When identifying organizational elements that work for or against a project a useful technique is

- a. SWOT analysis
- b. Decision tree
- c. Force field analysis
- d. Stakeholder identification

Q274. The process of allocating requirements deals with

- a. Maximizing business value and minimizing cost
- b. Assigning requirements to analysts for detail analysis
- c. Tracing requirements to product features
- d. Documenting which customer has what requirements

Q275. To help understand roles and responsibilities among stakeholders it helps to do a(n)

- a. Business case
- b. Process model
- c. Use case diagram
- d. Organizational model

Q276. To select the solution that best meets the requirements of the project, we would

- a. Assess the proposed solution.
- b. Assess organizational readiness.
- c. Do process modeling.
- d. Do solution modeling.

Q277. Jean has been part of the implementation of a new enterprise resource planning (ERP) system for her company. It was implemented 3 months ago, and Jean has been asked to go back and evaluate the solution performance. She quickly realizes that the users have implemented some workarounds to the system. How should she handle this?

- a. She should do nothing. The solution is implemented and is now owned by the users.
- b. Document lessons learned so these types of issues can be caught earlier next time.
- c. Assess the positive and negative effects of the workarounds.
- d. Escalate to the management of the functional area. ERP systems must be used consistently.

Q278. The finance department recently implemented a new system that just failed an audit due to lack of regulatory controls in the system. What would have helped them with this?

- a. SWOT analysis
- b. Force field analysis

- c. Process modeling
- d. Business rules analysis

Q279. When evaluating lessons learned from the order process improvement project, it was realized that there was a problem transferring work in progress from the old process to the new. This had not been identified as an issue from the start. What type of requirement was it that was missed?

- a. Transition requirement
- b. Nonfunctional requirement
- c. Solution requirement
- d. Business requirement

Q280. An operational assessment of a new solution can be useful to

- a. Assess problems in the current organization.
- b. Identify root causes.
- c. See if the organization has the capabilities to use the new solution.
- d. Establish metrics.

Q281. The level of involvement of the business analyst in the testing process varies from organization to organization. Which statement would normally be true?

- a. The business analyst tests the end product.
- b. The business analyst ensures that the product is tested.
- c. The business analyst validates that the product meets the business need.
- d. The business analyst is responsible for creating test plans.

Q282. What is the difference between *validate* and *verify*?

- a. Validate means to make sure we do the right things; verify means that we do it right.
- b. Validate is a process; verify is a product.
- c. They are the same.
- d. Validate is done outside of the project team; verify inside the project team.

Q283. Which is the best definition of solution assessment and validation?

- a. To verify that the solution meets requirements
- b. To ensure that the solution meets the business need and facilitates successful implementation
- c. To ensure that the business case is realized
- d. To assess and evaluate the business analysis process

Q284. Which of these would normally NOT be a solution component?

- a. People
- b. Business processes

- c. Software
- d. Development process

Q285. In order to assess if a solution, even though proven successful in one part of the organization, will prove successful in a new environment, the business analyst should perform a

- a. Cultural assessment
- b. Operational assessment
- c. Geographical assessment
- d. Solution assessment

Q286. The decision to automate a solution or handle it manually is part of which process?

- a. Defining the business need
- b. Documenting the project scope
- c. Allocating requirements
- d. Assessing the solution

Q287. If there is no good way to validate that a solution meets the business need, then

- a. The effort should stop.
- b. Analysis should start in order to define metrics and change the solution to allow for the capture of those metrics.
- c. The sponsor should decide if the project should be continued.
- d. The business need must be defined more.

Q288. Which of the following techniques may work well to assess the value of a solution to a group of stakeholders?

- a. Requirements workshop
- b. Focus group
- c. Brainstorming
- d. Job shadowing

Q289. The decision to run an old and a new system in parallel is a

- a. Business requirement
- b. Transition requirement
- c. Stakeholder requirement
- d. Solution requirement

Q290. When defining user classes, which is most important?

- a. Location
- b. User attitude
- c. Education
- d. Seniority

Q291. What is meant by acceptance criteria?

- a. Items that will be tested in the UAT.
- b. Items that must be in place to satisfy the business case.
- c. The successful end of a use case.
- d. Minimal requirements that should be met to make implementation worthwhile.

Q292. After a force field analysis is done, one should

- a. Strengthen the forces that support change.
- b. Get stakeholder approval.
- c. Assess strength of force.
- d. Decide if the project should continue.

Q293. A unique aspect of transition requirements is

- a. They are defined by the technical organization.
- b. They do not involve the users.
- c. They are not relevant once the current solution is eliminated.
- d. They are not traceable back to business objectives.

Q294. To complete a transition there should be a definition of

- a. Acceptance criteria
- b. As-is and to-be data models
- c. User acceptance test
- d. Sign-off criteria

Q295. Risk assessment during evaluation of organizational readiness

- a. Means determining potential problems and mitigation strategies
- b. Is the basis for a go/no go decision to implement
- c. Is to create implementation plans
- d. Is not solution focused

Q296. The following elicitation technique can be used to effectively identify problems with a solution after implementation:

- a. Workshops
- b. Document analysis
- c. Observation
- d. Benchmarking

Q297. What would be an example of nonfinancial costs of a solution?

- a. Increased workload during implementation
- b. Loss of corporate reputation
- c. Penalties for nonperformance
- d. Legal costs

Q298. Which of the following techniques could best be used to identify differences in to-be versus as-is at a business process level?

- a. Data flow diagrams
- b. Entity relationship diagrams
- c. Use cases
- d. State diagrams

Q299. The new IT solution will be implemented across the globe for XYZ Corporation. The infrastructure varies quite a bit from location to location. With what element does the business analyst need to be extra careful?

- a. Defining transition requirements
- b. Solution scope
- c. Cultural assessment
- d. Technical assessment

Q300. When allocating requirements, which of the following techniques is useful to break down the solution scope into smaller components?

- a. Activity diagram
- b. Entity relationship diagram
- c. Functional decomposition
- d. Use case

ACTIVITY SOLUTIONS FOR SWEDE-MART CASE STUDY

Chapter 1

The most common answers are

1. Changing requirements
2. Customers not available
3. Customers not knowing what they want
4. No repeatable process
5. Lack of management buy-in
6. Lack of resources
7. Lack of skilled business analysts
8. No common language
9. Developers do not understand the business
10. No tools

Actions

- Changing requirements
 - Implement a formal change control process
 - Educate team on the process
 - Customers not available
 - Give customers as much lead time as possible
 - Get buy-in from customers to requirements gathering
 - Customers not knowing what they want
 - Showcase similar products
 - Create prototypes
 - No repeatable process
 - Define and implement templates and a simple process to start with
 - Research what other organizations do
 - Lack of management buy-in
 - Review charter and scope with management
 - Review requirements plan with management
-

Meeting

This can be the start of what the Software Engineering Institute (SEI) calls a “Software Engineering Process Group,” which would become the owner of the processes adopted.

Chapter 2

ORGANIZATION	STANDARD	PROCESS OWNER	IN USE SINCE
IT	System specification	IT	At least 5 years
Finance	Project justification form	Director of finance	1990
Steering committee	Project charter	Not sure	Five years
IT	Enhancement request form	Director of IT	1990

Note that this will provide a starting point of forms and templates for the organization. It will also identify potentially overlapping forms and activities and help clarify roles and responsibilities.

Chapter 4

1. Risks and mitigation strategies

APPROACH	RISKS	MITIGATIONS
Waterfall	<ul style="list-style-type: none"> • Customer is not clear on requirements • Organization has not done anything similar in the past • Must be international consensus 	<ul style="list-style-type: none"> • Verify with prototype • Formal review of all deliverables • Add outside expertise team
Iterative	<ul style="list-style-type: none"> • Scope continuously changing • Can get analysis paralysis 	<ul style="list-style-type: none"> • Clearly define purpose and deliverables for each iteration
Agile	<ul style="list-style-type: none"> • Requirements continually changing • May be difficult to subdivide project into manageable components 	<ul style="list-style-type: none"> • Clearly define the boundaries of the effort • Identify pieces that can be done agile and combine with a more traditional approach for larger modules

Recommendation: On the basis of the complexity and size of this project, a waterfall approach would be the best option. Portions of the requirements gathering can be done in an iterative approach.

2. Requirements plan

a. Project overview and background.

Swede-Mart is expanding globally, and as it moves into new markets, it will have an increasing need to control the flow of materials. This project will replace the existing inventory and order system with a state-of-the-art model that will maximize product availability while minimizing cost.

b. Scope and deliverables: Identify three examples of exclusions.

The scope of the analysis phase include

- Create a plan for the analysis phase and obtain approval
- Collect requirements from the stakeholders

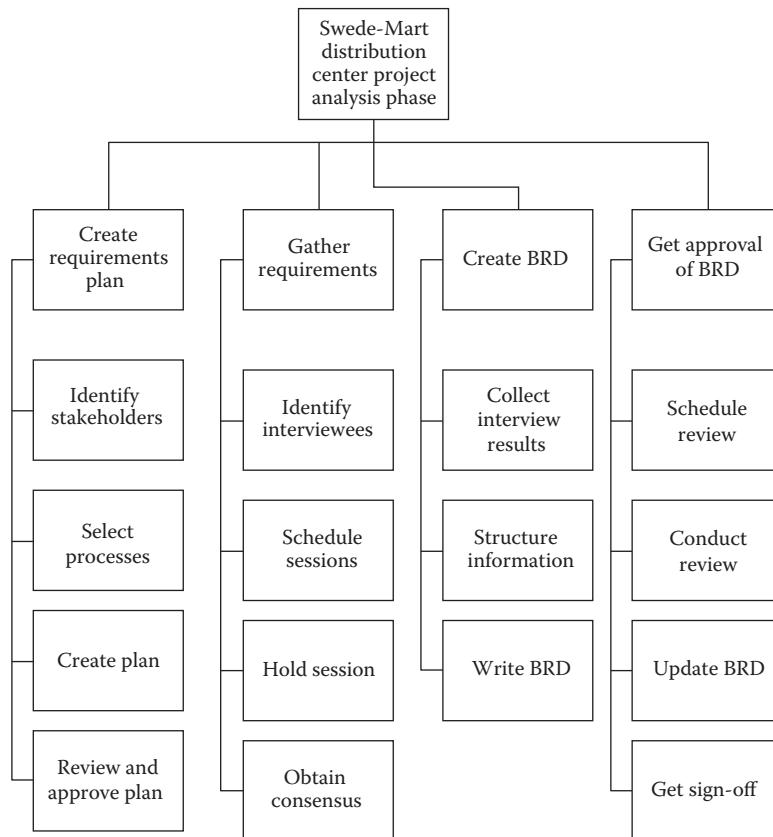
- iii. Create a business requirements document (BRD)
 - iv. Validate and get approval of the BRD
- The following areas are excluded:
- i. Requirements related to multilingual support
 - ii. User training
 - iii. Test cases
- c. Stakeholder analysis: Fill out the stakeholder analysis for Joe Jones.

STAKEHOLDER ANALYSIS	
Project name: Swede-Mart Inventory Project	Date: July 1, 2007
Project manager: Sven Svenson	Project sponsor: Joe Jones
Stakeholder organization:	Distribution centers
Stakeholder name and contact information:	Joe Jones (121) 555-1212
What will this stakeholder provide to the project?	Approve changes to the cost and schedule of the analysis phase Escalation point to determine the correct stakeholders in the requirements sessions
What will the project provide to this stakeholder?	Status reports BRD Product
What is the impact to this stakeholder if the project succeeds or fails?	Very important for the success of the stakeholder's organization as well as for his personal career
Hot issues for this stakeholder	Bad news early No surprises Bring ideas not problems Frequent communication

- d. Communications plan: Fill out with two sample communication items.

WHAT	WHO (RESPONSIBLE)	WHO (AUDIENCE)	WHY	WHEN	WHERE	HOW
Weekly status report	Lead analysts	Sponsor, project manager	Informational, continued support	Monday a.m.	Joe's office	In person
Team meeting	Project manager	Project team	Status, issues, upcoming milestones	Friday a.m.	Team room	In person, conference call

- e. Project activities: Create a work breakdown structure (WBS) with three layers (the top layer being the analysis phase) and a total of fifteen to eighteen activities at the lowest level.
- f. Roles and responsibilities: Assign responsibilities to the activities from Activity (e).



TASK	RESPONSIBLE
Create requirements plan	Lead analyst
Gather requirements	Business analysis team
Create BRD	Lead analyst
Get approval of BRD	Project manager, lead analyst, lead systems analyst

- g. Resource plan: Create a list of potential nonhuman resources and subject-matter experts you may need.

RESOURCE NEED	COMMENT
Facilitator	No internal candidate, must have superior communication skills
Documentation tool	Must support modeling techniques selected
Subject-matter expert	Need expert in inventory optimization

- h. Requirements risk plan: Brainstorm four to six risks. Select the top one and create a risk handling plan.

Risk Assessment Form (Type: Prod = Product Risk, Req = Requirements Gathering Risk)

RISK ID	TYPE	RISK	PROBABILITY	IMPACT	COMMENT
1	Prod	Local legal restriction may make the system not usable	L	H	
2	Prod	Hardware cost may be too much for some stores	L	M	
3	Prod	Some stores may not have English-speaking staff	M	H	
4	Req	May not be able to interview stakeholders from other countries	M	H	

Risk Handling Form

Risk ID: 4	Business area: Distribution centers
Raised by: Analysis team	Project name: Swede-Mart Distribution Center Project
Date raised: July 1, 2007	Assigned to: Lead analyst
Risk description: If we cannot interview users in other countries, we are likely to miss local legal restrictions and different nuances in how business is conducted.	
Likelihood: M	Impact: H
Impact description:	The acceptance of the product when rolling it out internationally could be jeopardized and the users may resist using the system, instead resorting to manual orders.
Recommended preventative actions:	Bring in a user from each continent to the requirements sessions. Send prototypes to each distribution center for review.
Recommended contingent actions:	Prepare to send trainers out at implementation time.

- i. Identify who will be responsible for approving changes.
 - There is a customer change control board that will approve product scope changes.
 - The customer sponsor will approve project scope changes.
 - The IT change control board will approve infrastructure changes.

Chapter 5**1. Elicitation**

STAKEHOLDER	ELICITATION APPROACH	SPECIAL ACTIONS
Joe Jones—sponsor Obtain sign-off on scope and requirements	Interview	Prepare list of questions
Buyer in store	Job shadowing 1 or 2 Survey the rest	Explain goals of project Create a survey for post-project evaluation
Distribution center inventory staff	Discovery session	Meet with each person before session Hire a JAD facilitator Select modeling techniques
Shipping	Focus group	Select participants who are well respected Brief participants before session Follow-up with decisions after the session

2. Discovery session for distribution center inventory staff

Attendees: Inventory managers from each continent
Inventory optimization
SME IT group supporting current inventory system

Agenda:

- Kickoff by Joe Jones
- Team-building activity
- Review goals and objectives
- Establish ground rules
- Review modeling techniques to be used
- Model the inventory flow
- Capture detail requirements
- Validate information captured
- Decide next steps

Chapter 6

Key stakeholders and their focus during BRD review:

- Sponsor: Scope, business requirements, assumptions, and constraints
- Distribution center manager: Business requirements and high-level user requirements
- Vice president of operations: Store requirements, assumptions and constraints, and use cases
- Buyer: Use cases, user requirements, and functional requirements
- Developer: All documents

Chapter 7

Project Charter

Project name: Swede-Mart Inventory	Customer: VP operations
Project manager: Sven Svenson	Project sponsor: distribution center manager
Business need/issue	With Swede-Mart's expansion globally and the focus on increasing market share and customer loyalty, a new order and inventory control system must be created. This will support Swede-Mart's vision of the right product at the right place at the right time.
Project justification	This market is increasingly competitive. Many competitors carry the same brands as we do. If we do not have them in stock when the customer needs them, the customer will go elsewhere and probably keep going there.
Critical success factors	<ul style="list-style-type: none"> • Never run out of inventory on year-round items • Less than 1 percent out of inventory of advertised products • Ability to tell customers when an out-of-inventory item will be available • Minimize cost of inventory
Key product deliverables and milestones	<ul style="list-style-type: none"> • Order processing system • Inventory management system

(Continued)

Organizational assumptions	<ul style="list-style-type: none"> • No other systems will change • Corporate language is English; no multilingual support
Organizational constraints	<ul style="list-style-type: none"> • Must be completed in 18 months • Must use existing infrastructure

Chapter 8

1. Sample requirements for the case study:
 - a. External: Swede-Mart must operate in accordance with local labor laws.
 - b. Business—strategic: Increase inventory turn rate to eight times each year.
 - c. Business—tactical: The distribution centers must be able to do daily trend analysis on inventory to forecast any increase (or decrease) in demand.
 - d. Business—operational: The store manager must be able to override default shipping in case of near out-of-inventory condition.
 - e. User: The buyer in the store must be able to place a special order for a product not carried in inventory.
 - f. Quality of service: A distribution center must be able to process orders from up to fifty stores each day.
 - g. Assumption: Users will be able to dedicate 4 hours to a training program for the system.
 - h. Constraint: All order history will be converted to the new system.
 - i. Implementation: The turnover from old to new system must be completed in 12 hours or less.

2a. Use cases

- Place special order
- Inquire on order status
- Inquire on inventory levels
- Order from nonprimary distribution center

Use Case: Place Special Order—General Characteristics

Intent	To place an order that was not automatically identified and generated by the system.
Scope	Store buyer placing special order
Author	Sven Svenson
Last update	July 15, 2007
Status	Under development
Primary actor	Store buyer
Secondary actor	Distribution center inventory manager
Precondition	System-generated order has been created and transmitted
Assumption	Buyer needs to modify system order
Trigger	Out of stock at store, special order at store, forecast changed
Success condition	Special order is confirmed for shipping with a ship date

(Continued)

Failed postcondition	Product cannot be ordered, no inventory available or special shipping cost prohibitive
Models Overview	Workflow model store purchasing process A buyer needs to try to make sure that the store does not run out of inventory. If the risk for this is high, they will need to place special orders. Special orders can also be placed when a customer orders an item that is not stocked.

Happy Path

STEP	ACTION
S	Successful order is placed for product in inventory at distribution center
1	The buyer decides to place special order
2	System asks for product
3	Buyer enters product
4	System displays available inventory
5	Buyer enters quantity
6	System confirms order and provides shipping date

Alternatives

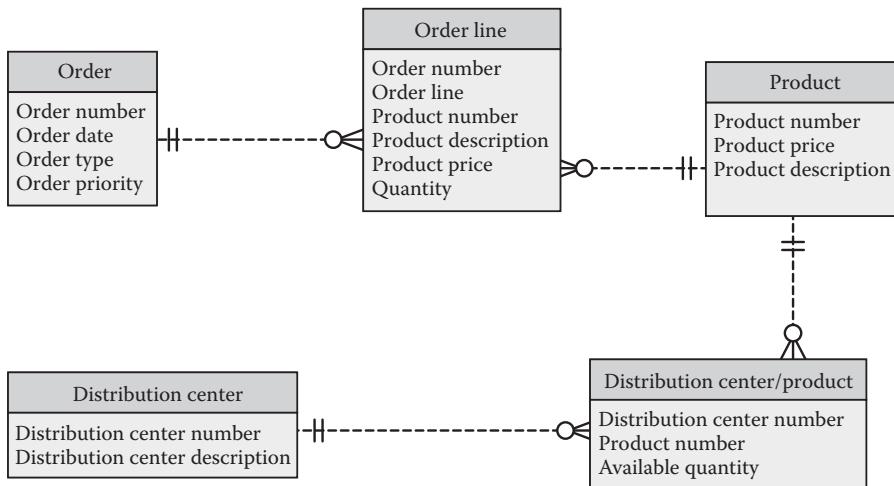
STEP	BRANCHING ACTION
4	No inventory available Buyer reviews all distribution centers Buyer evaluates shipping cost from closest distribution center with available inventory Buyer accepts or rejects alternative distribution center

Related Information

Performance	Must be able to perform happy path in less than 30 seconds
Frequency	Twenty times per day per buyer
Concurrency	Many buyers will be performing this concurrently
Open issues	Approval of orders from nonprimary distribution centers
Future considerations	None
Due date	2008
Additional information	N/A

2b. Entities

The entity relationship diagram is shown below.



2c. Requirement example

There will be many requirements statements for this use case. This is just one example: “The system must support fifty buyers per distribution center, all entering orders at the same time, and be able to complete those orders (assuming happy path) in 30 seconds.”

Chapter 9

1. Three parts of the case study where the customer is likely to have a very low risk tolerance are
 - a. Communication between store and distribution center
 - b. Accurate tracking of inventory levels
 - c. Any condition increasing out of stock on advertised products
2. Two test scenarios that would verify the requirement that the system must verify that the customer has sufficient credit to place an order, if the order is above \$1000, and if the customer does not have an existing line of credit with us:
 - a. Existing customer with line of credit placing an order for above \$1000
 - b. New customer placing an order for above \$1000



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

PREPARING FOR THE TEST

Ideally you are 2 to 6 months away from taking the test. The first thing you should do is a general review of all knowledge areas, including a sample test. That would give you an idea of where to focus your study. For instance, if you have worked with elicitation for a long time and are very comfortable with those techniques, you should focus more time outside the area of elicitation. However, if this is not your strong suit, study it hard, since it is normally a heavily tested topic.

Some people find that forming a study group is a great way to force some discipline into their study habits. However, the study group needs to be well organized, and often the biggest benefit comes from preparing presentations for the group. That forces some detailed research, which greatly enhances your knowledge of a topic. Your local International Institute of Business Analysis chapter may have some study groups.

Go from the general to the specific—make sure you know how the tasks interact and where the different knowledge area processes belong. By the time you have 3 to 4 weeks left before the exam, you should have completed your study of the general areas and be focused on the details that you need to work on. Many people find using quizzes and sample tests very useful during this time.

Some facts need to be memorized for the exam: data modeling terms, elicitation techniques, etc. Different approaches work best for different people. For some people this is easy, but for others this can be the major difficulty of the exam. One suggestion is to write down each point you want to memorize on a 3×5 card and carry it with you. That way you can quiz yourself without having to be in your study chamber. You can also get help from other people by having them quiz you using the cards. Children, nieces, nephews (and even some significant others) get a real kick out of helping you suffer through this, and it does add a little bit of pressure, which is a good simulation of the test.

The last 24 hours before the test can be major panic time. Be assured that no one ever feels as if they have prepared enough for the test. Some people say that you should not study the night before the exam, but that is personal preference. You should definitely plan on getting a good night's sleep. Try to keep to your normal evening and morning routine.

Give yourself enough time to get to the test site. The test is stressful enough; if you are already stressed out from traffic and trying to find the place, you will be at a disadvantage before you even start. Make sure you have identification and any items you need for the test; check this as you prepare for the test.

Taking the Test

There are 150 questions on the test. On the basis of the vagueness and ambiguity involved with a topic like this, it is unrealistic to expect to get everything right. People with a solid background in business analysis, with both practical and theoretical knowledge of the topic, should be able to get approximately 40 to 50 percent of the test without serious studying. The goal of preparing should be to get you up in the 80 percent range (leaving some safety margin). Most of this should come from increased skills, but some of it might come from *knowing how to play the game*.

The following is a list of test-taking techniques that have been found helpful in the past. Some are obvious (but often overlooked) and some are trickier. Naturally, if you have memorized the *Guide to the Business Analysis Body of Knowledge®* and all other reference materials, you should not need this, but for the rest of us

Answer ALL questions. There is no penalty for a wrong answer. If there are eight questions on the test that you are clueless about, guessing should, on average, give you two correct answers.

Use the elimination technique. Even though you may not know the answer, you may know which it is not. If you can eliminate one answer, your odds go to 33 percent; eliminate two answers and you have a 50/50 chance of getting one point.

Choose answers that sound upbeat and positive. Test makers are often trying to go for brownie points. If the question asks how important communication is, choose the answer that says “Key to the success of the project” rather than “Kind of important.”

Do not get too bogged down with one question. If you do not know the answer, mark it and come back to it later. You have 3.5 hours (210 minutes) for 150 questions. That is normally plenty of time. However, if you take 5 minutes each on the first ones, you might perceive that you are falling behind, which in turn can make you a bit stressed.

Do not focus too much on guessing or tracking your overall score. Some people try to guess how many correct answers they have, and track it through the test. It tends to just take away time from focusing on what is important—getting as many correct answers as possible.

ANSWERS TO TEST QUESTIONS

A1. (d) Business understanding and problem solving

Why? The person who is hired for this project will need to make sure that the business can operate with the new system, and there will be many problems that he or she has to handle. Knowledge of information technology will not be as critical for the business analyst on this project.

A2. (a) A capability needed to solve a problem

Why? This is the International Institute of Business Analysis (IIBA) definition of requirement. The other options are either objectives or solutions to the requirement.

A3. (b) Understanding how people, processes, and technology interact

Why? The business analyst must understand the business processes, the roles that people have in the organization, and how technology fits in.

A4. (b) A requirement is something that is needed in order to solve a problem. It should be focused on meeting an objective.

Why? Requirements must be put in the context of solving a problem, meeting an objective, or satisfying some type of rule or regulation. That is the starting point. After that, there are many characteristics of requirements. This goes back to the IIBA definition of a requirement.

A5. (d) What you had been talking to the customer about was the solution scope; in other words, the product description. You agree that you should not discuss project scope directly with the customer, unless you and the project manager have agreed on that ahead of time.

Why? The distinction between solution scope (sometimes called product scope or product description) and project scope is important. There needs to be cooperation between the project manager and the business analyst in these areas; however, in general, the business analyst documents the solution scope (or the product description) and the project manager documents how the solution will be developed (the project scope).

A6. (c) The business analyst should have done a better job documenting assumptions and expectations.

Why? Requirements are defined as a documented representation of a condition or capability. It may be a constraint, an assumption, an objective, or any other form of requirement, but it must always be written down.

A7. (c) Stakeholder requirements

Why? Stakeholder requirements focus on the needs of a specific stakeholder. Business requirements are enterprise objectives. Functional requirements are solution related. The IIBA does not have a category called enterprise requirements.

A8. (a) Nonfunctional requirements describe the characteristics of a system and the environment it needs to operate in.

Why? According to the IIBA, nonfunctional requirements describe the environmental condition under which the system must function and the qualities the system must have. Nonfunctional requirements are often vital to the success of the system and are often of vital importance to the business.

A9. (d) Business analyst: Interface with all relevant stakeholders to understand the policies and goals of the business. Work with others to help recommend solutions that will satisfy those goals.

Why? The business analyst is a liaison between business and development, but that is too limiting a description. He or she must truly understand what the business needs and, through working with others, recommend solutions to business issues.

A10. (b) This is an implementation requirement and needs to be submitted as a change control and approved by the change control board.

Why? This is a real requirement and should have been included in the requirements document. For traceability purposes, it must go through the formal change control process.

A11. (d) Message

Why? Noise is defined as factors that cause the message to be heard and interpreted differently than was intended by the sender.

A12. (a) Forcing

Why? Time is critical, so a decision must be made.

A13. (b) A stakeholder's need when interfacing with the solution

Why? The IIBA states "... needs that a given stakeholder has and how that stakeholder will interact with the solution."

A14. (c) It can be used when designing workshops to make sure the participants are up to the task.

Why? If the participants do not feel safe about the environment they are in, they are not likely to be creative. So a safe environment must be created.

A15. (c) Domain subject-matter expert is sometimes a role played by the business analyst.

Why? The business analyst is often familiar with the processes and problems in the domain being addressed by the project.

A16. (b) The business analyst is responsible for capturing the solution need.

Why? The project manager is responsible for capturing the project need and project scope. The business analyst is responsible for capturing the solution need. However, the solution scope is owned by the customer.

A17. (d) That the fact that we have spent money on a project does not mean it should continue.

Why? We must make decisions about what solution to select based on the merits of that solution, not based on what we have already spent.

A18. (a) A functional requirement

Why? It describes what the solution must do.

A19. (c) A transition requirement

Why? It deals with how to handle the transfer from the as-is to the to-be situation.

A20. (b) A stakeholder requirement

Why? It deals with what the stakeholder must be able to do.

A21. (a) Neutral

Why? To be effective as a facilitator, there must be trust from all parties, and that is difficult to achieve if the facilitator is not neutral.

A22. (c) May perform the tasks in any sequence

Why? The tasks in the *Guide to the Business Analysis Body of Knowledge® (BABOK®)* may be performed in succession, iteratively, or simultaneously.

A23.(a) Balancing solution scope and project scope

Why? There may be conflict between project needs and solution needs, and these conflicts must be handled and balanced.

A24.(b) Evaluate the flowcharts with other stakeholders and see if they meet the needs.

Why? The *BABOK* does not show all possible techniques, and other techniques may be considered as well.

A25.(d) It is determined what activities are needed in order to complete a business analysis effort.

Why? The purpose of this knowledge area is to plan the activities needed to gather, understand, and document requirements.

A26.(c) Accomplishes something of value

Why? A task should produce a demonstrable, positive outcome that is useful, specific, visible, and measurable.

A27. (c) They may have different states (verified, prioritized, etc.).

Why? The states will reflect what processes the requirements have gone through.

A28.(b) The solution

Why? Functional and nonfunctional requirements are part of the solution requirements.

A29.(b) A task can start while input is being developed.

Why? Input does not have to be complete; there just needs to be enough information for the task to start.

A30.(c) Needed or not needed

Why? All requirements should be needed.

A31. (b) Help structure a brainstorming session

Why? Mind mapping helps present a consolidated view of complex concepts and problems.

A32.(b) Describe the profession and domain of business analysis

Why? The BACCM is used to describe the business analysis profession and to communicate around it using a common terminology.

A33. (d) Change, need, solution, stakeholder, value, concept

Why? Based on the definition from *BABOK* v3.

A34. (c) Plan business analysis governance

Why? *Plan business analysis governance* identifies the parts of the business analysis effort that are needed to support the governance function.

A35. (a) Roles and permissions matrix

Why? The roles and permissions matrix is used to document what role is allowed to access what function in a system.

A36. (a) Data mining

Why? Data mining is used to find useful patterns and insights from data.

A37. (a) Visual thinking

Why? Visual thinking helps communicate complex concepts by using visual representation.

A38. (b) The onion diagram

Why? This helps identify different layers of stakeholders.

A39. (a) Root-cause analysis

Why? The technique helps with understanding of real problems and issues within the business.

A40. (b) Submit the change to the change control board.

Why? The business analyst is not a decision maker for requirements changes. That should be done by the change control board or whoever was designated in the project charter.

A41. (c) Do an icebreaking activity with the team to get them comfortable with each other.

Why? Inviting the sponsor to attend and reviewing roles and responsibilities are good ideas but may be less effective if the team members are not comfortable with each other.

A42. (c) Escalate it past your manager.

Why? Although this is not an easy choice, you must remain legally compliant, so option (b) is not a good choice. Options (a) and (d) are too rushed. You must first seek to solve the problem within your organization.

A43. (a) Elicitation risks

Why? That is the task at hand and most critical at this time. As you actually start to get requirements, you can look at other risk areas as well.

A44. (b) The project manager

Why? This is a subset of the project plan, and the project manager is the owner of that.

A45. (c) Frequent, short, informal communication

Why? With teams being co-located and small in size, this tends to work well.

A46. (c) Iterative

Why? This team is too disbursed and large for an agile development life cycle to work well. The system is likely to have a lot of requirements from different stakeholders, so development in iterations with multiple checkpoints appears best.

A47. (a) To minimize the chance of missed requirements

Why? A complete set of requirements is important for success, so care must be taken to identify potential expectations and requirements.

A48. (b) Templates or process

Why? Organizational process assets, such as tools and templates, give the business analyst a good starting point for the tasks that are to be accomplished.

A49. (d) Hold a kickoff meeting.

Why? A kickoff meeting can be held at each phase of the project and serves to get everyone going in the same direction and as a team-building activity.

A50. (c) Document what type of persons you need for the session and review with the sponsor.

Why? Always try to negotiate for resources but be prepared to do the best you can with the resources provided.

A51. (b) Is considered a change-driven approach.

Why? This approach focuses on gathering feedback regarding a working solution.

A52. (a) Decision analysis

Why? This is a good technique for evaluating the pros and cons of different options.

A53. (b) Describes organizational units and interactions with other internal and external units

Why? Enterprise architecture is useful for understanding who needs to be involved with initiatives.

A54. (d) Quality assurance

Why? RACI, brainstorming, and risk analysis are techniques that can help in stakeholder analysis.

A55. (c) Great! Who can I get more information from?

Why? A feasibility study is a perfect assignment for a business analyst.

A56. (c) It can guide the selection of methodology and processes for this initiative.

Why? Lessons learned from past projects is a great way to determine what we want to do differently in this initiative.

A57. (d) Co-located

Why? Frequent communication will be needed, and this is much easier if the team is co-located.

A58. (a) RFQ/RFI/RFP

Why? These activities can lead to legal issues with outside entities.

A59. (b) Structured walkthrough

Why? Make sure to include time in the plan for this.

A60. (c) Requirements attributes

Why? *Complexity, urgency, and source* are terms that provide information about the requirements.

A61. (a) Frequency of changes to requirements

Why? This relates most directly to the work of the business analyst. The others are more like to be project management measurements.

A62. (d) Impact of not doing the change

Why? To make a decision, the decision makers need to understand both the impact of doing the change and the impact of NOT doing the change.

A63. (b) Part of the project communication plan

Why? The business analyst's communication plan is a subset of the project communication plan, focusing on activities related to business analysis. It may be more detailed than the project communication plan.

A64. (b) In a change-driven approach, the requirements are gathered as late as possible.

Why? The focus is on short time cycles with clear deliverables, which means that requirements are gathered and implemented in a short time frame.

A65. (c) You plan near-term work in detail and things further out at a high level.

Why? There must be a high-level plan for the whole effort but detailed plans often cannot be made until preceding activities are complete.

A66. (b) Are the most involved with the solution.

Why? This will help designing communication.

A67. (a) Use cases

Why? Actors in use cases are likely to be stakeholders.

A68. (b) Number of interfacing processes and systems

Why? The more interfaces there are, the more complex the interactions among various stakeholders are likely to become.

A69. (d) Sarbanes–Oxley

Why? The other choices are related to enterprise architecture and process improvement initiatives.

A70. (c) It adds significant overhead to the business analysis work.

Why? Ensuring traceability takes time and effort.

A71. (d) Establishing the ground rules

Why? The intent of ground rules is to define acceptable behavior for the session. The parking lot is a good technique for handling out of context discussions, or discussions that must be escalated. Setting objectives may help keep the session on track. Taking a break may cool emotions. However, the best starting point is with good ground rules.

A72. (c) It may be difficult to get the people responsible for the interfacing systems to work with the project team.

Why? The effort under way may not be viewed as a priority for the owners of the interfacing systems.

A73. (b) Focus group

Why? A heterogeneous focus group can handle a large number of participants and allows people with different backgrounds to participate.

A74. (d) Creating a data dictionary and glossary

Why? It is important that everyone is working from the same definition of terms.

A75. (a) MoSCoW

Why? MoSCoW (Must, Should, Could, Won't) is helpful in determining what the customer must have and what can be delayed.

A76. (d) The survey questions are easy to understand.

Why? If the questions are not understood, the result will be useless.

A77. (c) Customers may have unrealistic expectations.

Why? Prototypes may look real and the customer sometimes may not understand why a prototype cannot be implemented immediately.

A78. (d) The group functions poorly; use an autocratic approach.

Why? In this stage, everyone is new and uncomfortable. The facilitator must push for interaction.

A79. (a) They are intuitive for users.

Why? A storyboard uses language the user can understand.

A80. (c) Clearly state the problem that is to be brainstormed about.

Why? It is difficult to keep a brainstorming session focused unless the objective is clear.

A81. (a) Interface analysis

Why? Interfaces to other systems are often overlooked by customers so the business analyst must take special care to include them.

A82. (b) Open ended

Why? Open-ended questions encourage broad answers and make interviewees comfortable because they can talk in more detail about their jobs.

A83. (d) Observation

Why? This allows you to see how users are actually doing their jobs currently.

A84. (c) A structured interview

Why? It will be important to plan the meeting and prepare questions in advance. This means there should be an agenda, and objectives should be clear for the meeting.

A85. (c) Benchmarking

Why? Benchmarking allows comparisons to strengths and weaknesses of other organizations.

A86. (a) Context diagram

Why? A context diagram shows a system or business area and its interfaces to the outside world.

A87. (c) Leveraging previous efforts

Why? This is the main advantage of document research.

A88. (b) Having an agenda

Why? The sessions must be planned and organized, which means that an agenda is always needed.

A89. (d) Surveys

Why? A survey can be sent out to everyone involved for them to complete at their locations.

A90. (c) A vendor assessment

Why? Buying something off the shelf means that the vendor must be evaluated for technical capability, financial stability, and other criteria set by the organization.

A91. (a) Decision tree

Why? A decision tree is a good method for organizing alternatives when a decision is to be made.

A92. (c) A neutral, experienced facilitator

Why? A neutral facilitator is needed to ensure that all sides are heard and that everyone feels comfortable to speak up.

A93. (a) Passive observation

Why? You should take good notes and observe the processes multiple times.

A94. (b) Preparing for the interview

Why? The business analyst should start by establishing goals and objectives for the interview.

A95. (d) Do an icebreaker activity.

Why? You should assess where the team stands in the forming/storming/norming/performing cycle of group functioning. If the team has not yet formed into a functioning group, then an icebreaker may help.

A96. (b) Communication plan

Why? The biggest difficulty on this project will be communication, so an effective communication strategy must be planned from the start.

A97. (d) Anonymous brainstorming

Why? Starting with each participant writing down his or her ideas and then posting them is a way to ensure that everyone is involved.

A98. (a) Forced-pair ranking

Why? Forced-pair ranking compares requirements to see which of the stakeholder's view is the most important.

A99. (c) Participants

Why? This is an output that results from preparing for elicitation.

A100. (b) Confirm results

Why? This validates that the documentation matches the stakeholders' understanding.

A101. (b) Explains the purpose of the observation.

Why? The users must understand that it is the function that is being observed, not them as individuals.

A102. (d) Realistic stakeholder expectations

Why? The stakeholders must understand that the prototype is a communications tool and not a solution ready for implementation.

A103. (a) Availability of information

Why? Accurate information about what a competitor does may be difficult to come by. However, if achieved, it may speed up the development process.

A104. (a) An agenda

Why? Having an agenda will help the facilitator keep the session on track.

A105. (a) Keep it simple

Why? Keeping the survey simple, with a few, easy to understand questions, is the best approach to take.

A106. (a) It can be good groundwork if the user has limited availability.

Why? It is good to be prepared and have some understanding of the current situation before meeting with users.

A107. (b) Customer

Why? The availability, preference, and location of the customer determine the elicitation techniques used.

A108. (c) It generates a lot of ideas in a short time.

Why? Brainstorming works best in the early stages when all ideas should be considered.

A109. (d) The boundaries of the interview

Why? It is important to stay focused, so it must be clear what is within the boundaries of the discussion and what is outside.

A110. (c) Does data modeling training with participants as needed.

Why? Data modeling is not always intuitive, so it is important that the participants understand the technique.

A111. (a) Multi-stakeholder project with disagreements expected

Why? Requirements workshops are designed to handle multiple stakeholders and to reach consensus.

A112. (c) Prioritizes

Why? Density dotting is a fast fun way of prioritizing requirements or features, often in less than 60 seconds.

A113. (d) To show the importance of the project.

Why? Having the sponsor kick off the session is a great way to show management commitment, which will motivate the team.

A114. (b) A focus group is not a decision-making group

Why? The purpose of the focus group is to elicit ideas. The requirements workshop normally decides or recommends requirements.

A115. (a) Let the team work; it is now in the performing stage.

Why? When the team is performing, the facilitator needs to resist the impulse to ruin the session by intervening unnecessarily. Just monitor the group and let the team do the work.

A116. (b) Parking lot

Why? The parking lot can be used for items that are outside of the scope of a session.

A117. (b) Have a face-to-face meeting before the workshop.

Why? The remaining sessions will be much easier if the team members know each other. If they have not met before the workshop begins, communication will be very difficult.

A118. (a) To evaluate technical feasibility.

Why? Evaluating technical feasibility would be the purpose of a design workshop, not a requirements workshop.

A119. (c) It has a different look and feel than the final product.

Why? Storyboarding is normally done quickly and may not be an accurate picture of the finished product.

A120. (d) Each iteration has its own goal

Why? It must be clear what the expected end result is of each iteration.

A121. (c) Learn for future projects.

Why? Observers should never participate in the session, nor do anything to stifle creativity.

A122. (a) Stakeholder availability and preference

Why? Elicitation efforts will most often be driven by the stakeholders.
How much time do they have? Where are they located?

A123. (d) Process flow

Why? The process can be documented while observing the work being done.

A124. (b) Iterative

Why? Eliciting requirements will be done at different levels in the organization and at different levels of detail.

A125. (b) Observation is an as-is technique.

Why? The main purpose is to allow the observer to see what is actually being done.

A126. (d) Let the team work it out as long as ground rules are followed.

Why? Conflict is part of the group process, especially during the storming phase but ground rules must be kept.

A127. (c) Interviews

Why? Interviews are preferred when stakeholders are in different locations and occupy higher levels in the organization.

A128. (a) Decisions and rational for decisions

Why? How much information is needed will vary from session to session, but there must be enough to understand the decisions and why they were made.

A129. (c) The project manager

Why? Unless predefined or decided in the session (which is the preference), the project manager would have responsibility for tracking issues. This could be delegated to the business analyst in the definition of roles and responsibilities.

A130. (c) Requirements workshop

Why? A requirements workshop normally is 1 or 2 days in duration, ideally with no more than seven people attending.

A131. (d) The business analyst should review the requirements using a process that ensures that the stakeholders understand the requirements.

Why? It is not enough to just hand out a document. The business analyst needs to ensure that the requirements are understood by the stakeholders.

A132. (b) The level of detail should be based on the needs of the stakeholder.

Why? Part of preparing a requirements package is to determine how detailed the communication needs to be for different stakeholders.

A133. (c) Expressing them in terms that make them usable in the long term.

Why? Requirements should be captured and documented with reuse in mind.

A134. (a) Coverage matrix

Why? A coverage matrix is useful for tracing features or high-level requirements.

A135. (a) Determine the relationships between the requirements.

Why? This is needed to know which requirements should be grouped together for prioritization based on their impact on other requirements.

A136. (c) Must go through formal change control.

Why? Requirements may be baselined. If so, all changes (if allowed) must go through a formal change control process to evaluate their impact and desirability.

A137. (b) Escalate to the sponsor.

Why? Some action must be taken to resolve the conflict. Meeting with the parties involved, going to a third party for resolution, or escalating are all acceptable options.

A138. (c) RFI (request for information)

Why? RFP and RFQ are done when the solution has been tentatively decided upon. RFS has no meaning in this context.

A139. (a) How well does the requirements package support the solution development?

Why? The main focus here is to communicate with the developer, so the communication needs to be focused on what they need in order to do their job.

A140. (b) Difficult to find and retrieve unless the process formalized

Why? Requirements reuse must be formalized, and there need to be resources to manage the process. If not, finding, retrieving, and using the requirements will be difficult.

A141. (d) Who has authority to sign what portion of the document.

Why? There are often different stakeholders approving different parts of the requirements document, and these stakeholders must be identified.

A142. (c) A deliverable used as a basis for solution development and implementation

Why? This is a formal *BABOK* deliverable. There are general deliverables and requirements deliverables.

A143. (c) Prioritization

Why? Necessity, effort, subset, cover, and value are defined by the IIBA as attributes to help prioritization.

A144. (b) Problem tracking

Why? This technique allows the business analysis to track any identified issues.

A145. (c) Formal change control

Why? Requirements are selected for each iteration and changes are not allowed.

A146. (b) Solution scope will drive large portions of the project scope.

Why? Solution scope will define the product developed within the project scope. Project scope is the responsibility of the project manager.

A147. (a) Easy-to-understand requirements because they will be directly impacted.

Why? The requirements must be in an easy-to-understand language, employing terms that are actually used by the users.

A148. (d) Make decisions regarding scope

Why? Stakeholders should be well prepared with handouts and research to enable decision making.

A149. (b) Sponsor

Why? The sponsor and senior executives would normally receive summary information.

A150. (c) At different stages depending on the stakeholders

Why? Sign-off can happen at different times based on stakeholder availability, and also the need to get approval of higher-level requirements before detail requirements are developed.

A151. (d) Forward traceability

Why? The technique of forward traceability shows where each requirement is allocated.

A152. (a) Grouping requirements for implementation

Why? The relationship *necessity* is used when it makes sense to implement a requirement only when a related requirement is also implemented.

A153. (b) Testers

Why? Documenting requirements for reuse is important for regression tests conducted by testers.

A154. (c) Throughout the process

Why? Used when communication about requirements is needed in a clear and understandable way.

A155. (a) Functional and nonfunctional requirements

Why? These must be tested in the solution.

A156. (c) Avoid closed-ended questions.

Why? The vendor should be encouraged to provide extensive information, and this is best accomplished with open-ended questions.

A157. (b) It should be done iteratively.

Why? This is done throughout the project in conjunction with most tasks.

A158. (b) Activity diagram

Why? The activity diagram can show activities and hand-off between business areas.

A159. (c) Requirements approach

Why? This can determine how requirements are gathered but does not show what the requirements are.

A160. (b) Notes and diagrams used during the requirements development process.

Why? These items may or may not become deliverables.

A161. (a) Context diagram

Why? A context diagram shows the boundaries between the product and the outside world, with the scope in the center. The other models are primarily to analyze requirements.

A162. (b) Review project charters

Why? The charters are used to define projects after the business needs have been identified. The other three techniques are all used to identify potential needs.

A163. (d) It defines an appropriate scope for the solution

Why? The enterprise analysis links the business problem to an appropriate solutions approach and determines the scope of that solution.

A164. (a) SWOT analysis

Why? SWOT (strengths–weaknesses–opportunities–threats) analysis is a good quick way to focus on the main points impacting the other business. While the other techniques are good techniques, they would tend to be more time consuming or used later in the process, when comparing alternatives.

A165. (b) Root-cause analysis

Why? The problem was stated but there did not seem to be a determination of what the cause of the problem was. A root-cause analysis would evaluate different potential causes and determine to most likely cause or causes.

A166. (c) Business case

Why? In order to develop a business case, the scope of the solution must be defined. The other three answers refer to items that should all have been created before the solution scope was created.

A167. (b) He is responsible for describing the business environment and the business requirements.

Why? While scope definition is primarily a project management task, there are sections in that document that are the responsibility of the business analyst, such as the description of the as-is situation and the high-level requirements of the solution.

A168. (d) Fishbone diagram

Why? This provides good structure and organization. The Five Why's is a good technique but can get a bit annoying for some customers. The other techniques are too free form for this customer.

A169. (b) Organize a Zachman framework model to understand the business as it is today.

Why? An enterprise framework will give a good solid understanding of all aspects of the business. With that in place, the CEO can start identifying problem areas and business opportunities, and do a gap analysis. Creating the Zachman framework is a major undertaking; however, in this case, there appears to be strong executive support for the effort, which is critical for it to be successful.

A170. (c) This could be really good. Business use cases help describe the key functions and services that the business provides for its stakeholders.

Why? Business use cases help describe the key processes with which a business is involved and identifies the actors that are involved with those

processes. This can be a good part of the business architecture definition. There are other ways to model this so the organization must evaluate what best fits its organizational skills and experiences. It is always good to set a standard for the organization because business use cases, like all other models, take time to learn.

A171. (a) Determine the gap that exists in the organization as far as dealing with this business need is concerned.

Why? Strategic goals should be known before the business need is determined. The recommended solution and the business case cannot be determined until you know the gap in your current organizational abilities.

A172. (d) Solution scope

Why? This needs to be an input into creating the business case. The outputs are costs, benefits, initial risk assessment, and a process for measuring results.

A173. (a) A conceptual data model would be very useful in understanding the business.

Why? Information plays a key role in any business and the conceptual data model focuses on showing key information without going into unnecessary levels of detail.

A174. (c) Business goals

Why? The business architecture describes the business, including its vision, goals, organization, geography, etc. The technical architecture describes tools, applications, software, hardware configuration, etc.

A175. (d) Preliminary study of solution options and business benefits

Why? A feasibility study addresses solutions to business problems or opportunities to see what value they bring.

A176. (b) Description, who is affected, and who is impacted

Why? The focus should be on first understanding what is happening.

A177. (d) Functional requirements

Why? These will typically be developed after the business case is done.

A178. (a) Change the business procedure

Why? The other three are activities that are undertaken after the solution approach is done.

A179. (a) Impacted organizations

Why? Goals, assumptions, constraints, and impacted areas must be defined.

A180. (c) The gap between the current situation and the to-be situation.

Why? Gap analysis focuses on what is needed to go from today's situation to the future situation.

A181. (c) Qualitative and quantitative evaluations of the benefits of a project.

Why? The business case should provide as much information as possible about costs and benefits in order to facilitate a good decision.

A182. (c) Project manager risks

Why? The initial business case is typically done before the project is approved and started. This means that the project manager is not yet assigned.

A183. (a) A product recall

Why? The other options are responses to events.

A184. (d) A requirement cannot contradict a business rule.

Why? The business must be run according to its business rules. Enforcing the rules is not the purpose of a solution (they can be manually enforced). The solution should never contradict the business rules.

A185. (b) A feasibility study

Why? A feasibility study is a preliminary analysis of solution options.

A186. (a) Solution scope

Why? Stakeholders must understand the boundaries and impact of a solution.

A187. (b) Infrastructure model

Why? The infrastructure model shows the information capability of an organization by identifying the network, computers, and servers at different locations.

A188. (a) For modeling interactions between businesses.

Why? The business entity model shows flow between vendors, customers, and one's own business.

A189. (c) Tracing the business need to the solution.

Why? This step is done during solution assessment and validation, not enterprise analysis.

A190. (b) To identify and define why we need to change.

Why? Stakeholders must know the reason for change at the start.

A191. (a) Implementing in multiple releases.

Why? The approach to implementation defines how the solution is implemented, not developed.

A192. (d) User stories

Why? User stories identify stakeholders and their goals.

A193. (d) Benchmarking

Why? Benchmarking works best when there is data available from organizations regarded as best in class.

A194. (b) Zachman framework

Why? The Zachman framework shows the what, who, where, when, how, and why of a business.

A195. (c) Look at the impact on each stakeholder.

Why? Some may be positive; some may be negative.

A196. (a) Capitalizing on strengths.

Why? Capitalizing on the organization's strengths is likely to give it a competitive advantage.

A197. (c) Staff competencies

Why? Assessing the capability gap should focus on what the as-is situation is and what is needed to be successful in the to-be situation.

A198. (d) Defining the business need.

Why? Root-cause analysis is part of understanding the business need.

A199. (a) To evaluate if the initiative is riskier than the organization is willing to tolerate.

Why? The initial risk assessment is part of the recommendation that goes with the business case.

A200. (d) Comparing all revenues and costs based on today's value of money

Why? The net present value is the current monetary value of revenue and cost based on the cost of capital.

A201. (b) The BCR is 2.

Why? BCR = the benefit/cost ratio. There is not enough information given to determine net present value (NPV), payback period, or internal rate of return (IRR).

A202. (b) Any solution must use existing infrastructure.

Why? Constraint limits solution options. The other choices are phrased as assumptions or objectives.

A203. (c) Analogous

Why? The analogous technique is useful for high-level estimating and comparing with earlier efforts. It is used when not much detail is available.

A204. (a) Develop a business case

Why? This is the most appropriate response; the other choices refer to processes that come later.

A205. (a) The time it takes to get the investment back

Why? This is the time when total investment = total benefit.

A206. (d) A focus group

Why? This would be good for a broad assessment of potential problems.

A207. (c) Decision making

Why? The analytical hierarchy is used for comparing different factors and scoring them, which facilitates good decision making.

A208. (c) Include information about how they are enforced.

Why? Business rules should be independent of the solution.

A209. (b) Although the business analyst provides input into cash flow analysis, it is not his or her primary concern.

Why? Cash flow analysis is usually the responsibility of someone higher up in the organization.

A210. (a) Document analysis

Why? A good practice in elicitation is to start by seeing what has already been done.

A211. (c) The shareholders

Why? Anyone or anything that initiates a use case is a primary actor. It can be a person, a system, or something more abstract such as time (*process annual tax statements* would be an example). While the shareholders would be stakeholders, they would normally not be direct users of the system.

A212. (a) Good catch. We will submit it as a change control to the level 0 and level 1 DFD and evaluate the impact.

Why? Data flow diagrams must be balanced between the levels. All external data flows into or out of any level of the diagram must show up on a higher

level. If one is missed, there needs to be an evaluation of impact on other areas of the model.

A213. (d) The order can only be placed by one customer.

Why? The relationship from customer to order is zero or many, meaning that while the customer *may* place many orders, it is *optional*, meaning that someone can be a customer even without having placed an order (a potential customer). The relationship from order to customer is one and only one. This means that every order belongs to a customer and no order can belong to more than one customer.

A214. (a) The customer places zero or many order items.

Why? Only option (a) is correct because a customer may place no order but, if she does, may place many orders (with potentially many order items).

A215. (d) Change the cardinality from the order to the customer to be zero or one.

Why? This will allow the customer field to be optional. At some point in the process, the customer must be added. This can be dealt with by a process rule. Note that in the physical model, option (a) may be OK, but it does not belong in a logical model. In the logical model, we describe that data as the customer sees it, and from that viewpoint, the customer is optional in this case.

A216. (c) A diagram and a text document

Why? The diagram gives the visual representation and the text document gives the explanation and detail.

A217. (c) Validating requirements

Why? Verifying means ascertaining that the solution met the requirements, which it did. Root-cause analysis only states that the real problem was identified, which is not clear from the question. Validating means that the solution addresses the real business problem, which clearly describes the problem here. This is definitely a process that the business analyst should be involved with.

A218. (b) That an iterative approach is often appropriate, with each iteration elaborating the requirements further.

Why? Progressive elaboration is focused on first capturing requirements at a high level and then gradually elaborating into more detail. It is another term describing an iterative approach to requirements gathering.

A219. (d) The conceptual data model is used to communicate with the customer and the logical data model is used to communicate with the developer.

Why? The main difference between the models is the audience. The level of detail is often higher in the conceptual data model, but the reason to use it is to communicate at the level with which the customer is comfortable.

A220. (b) You agree. The organization's investment in the tool requires a commitment from all areas of the business.

Why? Although you must make sure that the customer relates to the techniques you are using, the investment in the tool makes it important that you at least try to use the recommended methodology. Use cases should be intuitive for the customer.

A221. (a) Writing independent requirements that are not connected to other requirements.

Why? The requirements statements should be simple and understandable. They will often be dependent on other requirements.

A222. (c) Implementation requirement

Why? Implementation requirements define what must be done to get ready for the product. Data conversion and training fall in this category.

A223. (d) Validating requirements

Why? To validate means to make sure that all the requirements trace back to the business problem, goals, and objectives. Verification means that the format meets the organizational standards.

A224. (d) You make a note in the business requirements document but do nothing.

Why? This is the responsibility of the solution developer but you should always document ideas and potential solutions so the solution developer can know what has been discussed.

A225. (b) Event modeling

Why? Event modeling would help identify specific business events that impact the organization and which may need to be dealt with by the project team.

A226. (c) Put them in an *extends* use case to extend the functionality later.

Why? Extends are used to model alternative scenarios in a use case. This is often related to implementation considerations.

A227. (d) Functional decomposition diagram

Why? The class diagram and the data model do not really deal with business processes. Use cases may help, but they are focused on what an external actor wants to do with the system. A functional decomposition diagram would give a good overview of the business's functions and processes.

A228. (c) Ownership of data

Why? A CRUD matrix shows who can create, read, update, or delete data.

A229. (b) We can verify that the requirement is met after the solution is done.

Why? If there is no way to verify that a requirement has been implemented, then the requirement is useless.

A230. (a) State diagram

Why? A state diagram shows what events bring an entity from one state to another.

A231. (c) In a nonfunctional requirements document

Why? These requirements often are the same across many use cases, and therefore they are best captured in a nonfunctional requirements document.

A232. (a) Storyboards

Why? This will be the fastest method and will also be intuitive for different stakeholders.

A233. (d) Business rule

Why? A business rule guides how the business is run. This may also become a requirement if the solution must enforce this business rule, but at this time this is not clear. This could be a manually handled process. It becomes a requirement after it is prioritized and approved for the solution.

A234. (d) Sequence diagram

Why? A sequence diagram describes when and how messages are sent between objects.

A235. (c) A class diagram shows operations on the data; ERD does not.

Why? A class diagram shows the class, attributes, and operations on the data. The entity relationship diagram (ERD) does not show operations.

A236. (a) What state the system is in at the end of the use case

Why? Precondition shows what state the system is in when the use case starts, and postcondition shows what state the system is in when the use case has been completed.

A237. (b) Functional decomposition diagram

Why? The functional decomposition diagram is a traditional diagram showing processes. It works with the entity relationship diagram showing data and the data flow diagram showing flow between processes.

A238. (b) Only one start point

Why? The other choices are not appropriate: it is possible to have multiple ends; the system can have its own swim lane; and loops are allowed.

A239. (c) Structure, behavior, interaction

Why? There are many models within UML but these are the three main groupings.

A240. (a) Communication

Why? Models aid in communication with customers, developers, and other stakeholders.

A241. (d) A model that is used for communication with the customer

Why? The conceptual data model may not show all entities, relationships, and attributes. It is focused on customer communication.

A242. (a) It collects data for some time period.

Why? A data store is a user view of data. It can be a paid invoice, a pending payment, or anything that the user views as a grouping of information stored for some time.

A243. (c) Entities, attributes, relationships

Why? These are the main components of an entity relationship diagram. It does not show operations or methods. Cardinality is a part of the relationship.

A244. (d) Two business rules

Why? The relationship shows business rules for how entities are viewed. There is one business rule in each direction for the relationship.

A245. (b) Take an external view of the system.

Why? Traditional modeling looks at the internal workings of the system, while use cases look at the process from the user's viewpoint.

A246. (a) State diagram

Why? State machine diagram, state transition diagram, and entity life cycle diagram are different names for a state diagram.

A247. (b) As an asynchronous flow in a sequence diagram

Why? An asynchronous flow allows the object to continue processing after the signal is sent.

A248. (c) Description, benefits

Why? User stories are short descriptions from the user view and include goals and any information needed to understand the scope.

A249. (d) As a constraint

Why? This is something that constrains your solution options. It could be considered a nonfunctional requirement, but constraint would be a better place to put it.

A250. (b) When working with executives, use a top-down approach; with end users, create a bottom-up view.

Why? Select the method that reflects how the stakeholder views the business.

A251. (a) *Car* and *Truck* will inherit all attributes and relationships from *Vehicle*.

Why? A subtype always inherits everything from the supertype but can also have its own attributes.

A252. (c) That a test case can be designed

Why? Since the solution is not yet done, the actual design of the test case may need to wait, but we must see that the test case is designable.

A253. (b) Business requirements

Why? Business requirements, which are documented as goals and objectives, must be aligned with the more detailed requirements such as stakeholder, solution, and transition requirements.

A254. (d) All of the above

Why? The primary actor is the person or system that initiates the use case. All other actors, people, or systems are secondary.

A255. (b) An event triggered by the passage of time

Why? Examples may be month-end processing, end of tax year, time to place a stock order.

A256. (a) Keep equality at the borders

Why? This means that all external entities from the parent level must be reflected on the child level and vice versa.

A257. (d) All types of interfaces

Why? User, hardware, and application interfaces are all covered by this.

A258. (b) Inputs and outputs

Why? *No inputs* means we have a miracle; *no outputs* means there is a black hole.

A259. (c) Data flow diagram

Why? The customer normally relates well since this is describing the business.

A260. (a) Yes; the first statement would allow customers who have never placed an order to do so; the second would not.

Why? The word *may* means optional, and this would lead to a different system design so this is important. It is not a business analyst's job to decide what is right, just to push for clarity.

A261. (b) Evolutionary prototype

Why? It evolves into the final solution.

A262. (c) An event

Why? An event takes an object from one state to another.

A263. (b) It normally requires a fully defined class model.

Why? That is the formal approach.

A264. (d) One way for the actor to meet the goal.

Why? A use case shows all scenarios.

A265. (a) Process model

Why? The process model shows sequence, flow, and interactions.

A266. (c) Updating information

Why? Updating information is a functional requirement.

A267. (c) A simplified representation of a complex reality

Why? This is the best answer because a model can have both text and graphic elements.

A268. (b) Assume that the reader has basic domain knowledge.

Why? The reader may be a developer without domain knowledge.

A269. (a) Negotiation

Why? You will need to negotiate for resources and support from the other organizations.

A270. (d) At least one relationship

Why? Each relationship describes two business rules. Related entities have at least one relationship but could have more (*customer pays for an order* and *customer places an order* shows two relationships between two entities).

A271. (b) Opportunity cost

Why? Organizations have limited resources and must evaluate what cannot be done (other opportunities) if resources are spent on this project.

A272. (a) The solution meets the business need.

Why? Validation is focused on making sure that the business problem or need is being addressed.

A273. (c) Force field analysis

Why? Force field analysis identifies forces for and against a change.

A274. (a) Maximizing business value and minimizing cost.

Why? Allocating requirements involves identifying as early as possible features that will create the most value relative to the cost.

A275. (d) Organizational model

Why? An organizational model describes the roles, responsibilities, and reporting structure within an organization.

A276. (a) Assess the proposed solution.

Why? Assessing the proposed solution involves evaluating how well it matches requirements.

A277. (c) Assess the positive and negative effects of the workarounds.

Why? Workarounds are normal and often needed in an organization.

A278. (d) Business rules analysis.

Why? Analysis of the business rules would have defined the organizational rules, constraints, and operations and helped to prevent implementation of a system that did not take these into account.

A279. (a) Transition requirement

Why? This describes what is needed to ensure a smooth transition when implementing a new solution.

A280. (c) See if the organization has the capabilities to use the solution.

Why? This is part of assessing organizational readiness.

A281. (c) The business analyst validates that the product meets the business need.

Why? The other roles described may be filled by the project manager, quality control manager, or quality assurance manager.

A282. (a) Validate means to make sure we do the right things; verify means that we do it right.

Why? Validation traces requirements to business need; verification traces product to requirements.

A283. (b) To ensure that the solution meets the business need and facilitates successful implementation.

Why? This is the IIBA definition.

A284. (d) Development process

Why? Solution components can be software, people who operate the solution, policies, processes, etc.

A285. (a) Cultural assessment

Why? The business analyst will assess the culture of the organization to assess the likelihood that the solution will be accepted.

A286. (c) Allocating requirements

Why? This process will allocate requirements between solution components (which may include manual processes).

A287. (b) Analysis should start in order to define metrics and change the solution to allow for the capture of those metrics.

Why? The validation of the solution must be aligned with the business need.

A288. (b) Focus group

Why? The focus group can be a forum for presenting solutions and asking for ideas.

A289. (b) Transition requirement

Why? A transition requirement deals with moving from the as-is situation to the to-be situation.

A290. (a) Location

Why? Although the other choices could play a role in defining users, location is often key to classifying users.

A291. (d) Minimal requirements that should be met to make implementation worthwhile

Why? Acceptance criteria determine go/no go decisions.

A292. (a) Strengthen the forces that support change.

Why? This is the best way to improve the chances of successful implementation.

A293. (c) They are not relevant once the current solution is eliminated.

Why? Transition requirements show how to go from the current solution to the new solution, and therefore are no longer necessary when the transition is complete.

A294. (b) As-is and to-be data models

Why? It must be possible to map information from the old solution to the new solution.

A295. (a) Means determining potential problems and mitigation strategies.

Why? Risk assessment identifies problems relating to the readiness of the organization for the change.

A296. (c) Observation

Why? Observation can help evaluate how the solution actually works after implementation.

A297. (b) Loss of corporate reputation

Why? Although corporate reputation is difficult to quantify in dollar terms, it could play an important role in decision making.

A298. (a) Data flow diagrams

Why? Data flow diagrams show business processes with inputs and outputs.

A299. (d) Technical assessment

Why? This is part of assessing organizational readiness.

A300. (c) Functional decomposition

Why? Decomposes solution scope into smaller components.

Appendix A: Acronyms

AHP	Analytical hierarchy process
BABOK®	<i>Guide to the Business Analysis Body of Knowledge®</i> (from the International Institute of Business Analysis)
BRD	Business requirements document
CASE	Computer-aided systems engineering
CMMI™	Software Engineering Institute's Capability Maturity Model Integrated
COTS	Commercial off-the-shelf software
CRUD	Create/read/update/delete
DFD	Data flow diagram
DSDM	Dynamic systems development methodology
ERD	Entity relationship diagram
ERP	Enterprise resource planning
FDD	Functional decomposition diagram
I-CASE	Integrated CASE
IIBA®	International Institute of Business Analysis®
IPO	Input/process/output model
ISO	International Organization for Standardization
IT	Information technology
JAD	Joint Application Development (or Joint Application Design)
JIT	Just-in-Time
OMG	Object management group
OO	Object oriented
OPM3	Project Management Institute's Organizational Project Management Maturity Model
PMBOK™	<i>Guide to the Project Management Institute's Body of Knowledge™, Fifth Edition</i>

PMI™	Project Management Institute™
RAD	Rapid Applications Development
RUP™	Rational Unified Process™
SDLC	Systems development life cycle
SEI	Software Engineering Institute (from Carnegie Mellon University)
SME	Subject-matter expert
SOW	Statement of work
SWAT	Skilled with advanced tools
SWOT	Strengths–weaknesses–opportunities–threats
UML	Unified Modeling Language
WBS	Work breakdown structure
XP	Extreme Programming

Appendix B: Business Requirements Document Templates

The following templates have been included as reference material. All templates should be customized to fit the organizational and project environments in which they will be used.

- B1. Business Requirements Document (BRD) Comprehensive (Courtesy of ESI International)
- B2. Business Requirements Document (BRD) Simple

**B1. Business Requirements Document (BRD) Comprehensive
(Courtesy of ESI International)**

This document is used by ESI International in its Business Analysis Curriculum, and ESI has kindly allowed it to be included in this book.

Business Requirements Document for [insert Project Name here]

Prepared by

Prepared for

Date submitted

Project sponsor

Client acceptor

Project manager

Business analyst

File name

Document number

Last edit

Comments

Important notice: This template includes instructional summaries at the beginning of each section. They use a style called instructions, which has been configured as hidden text. This means you can choose to display and print them, at will. Under normal circumstances, these should be hidden when you distribute your actual reports, unless you judge that the instructions themselves provide value to your readers. Please see the last section in this template for step-by-step instructions on how to see and print hidden text, along with shortcut keys for the various styles used in the template.

Table of Contents

Section Zero: Positioning of the Business Requirements Document	338
Goal: Common Understanding through Structured Business Analysis and a Standard Business Requirements Document.....	338
Word of Caution about Removing/Adding Sections	339
Different Types of Requirements	339
Prioritizing Requirements	340
Section One: Glossary.....	340
Section Two: Project Scope and Objectives Summary	340
Section Three: Technology Infrastructure and Information Architecture Compliance	341
Section Four: Intended Audience.....	341
Section Five: Decision Making and Approval Process for the Business Requirements Document.....	341
Section Six: Approach	342
Section 6.1—Overall Project Management Approach	342
Section 6.2—Business Analysis Approach	342
Section Seven: Background, Historical, and Prior Project Information.....	342
Section Eight: Business-Level Requirements: Goals, Value Proposition, and Benefits.....	342
Section 8.1—Regulatory Requirements	343
Section 8.2—Related Strategic Goals (Organization Level).....	343
Section 8.3—Related Tactical Goals (Division or Department Level)	343
Section 8.4—Related Operational Goals (Staff Level)	343
Section Nine: User Class Profiles and Key Delegations	343
Section 9.1—Sponsors and Stakeholders	344
Section 9.2—Primary Users	344
Section 9.3—Secondary Users	344
Section Ten: User and Functional Level Requirements	344
Section Eleven: Additional Information regarding Functional Requirements Related to Output and Reporting	345
Section Twelve: Conceptual Data Model	345
Section Thirteen: Nonfunctional Requirements	345
Section 13.1—Operational Environment.....	345
Section 13.2—User Interface Requirements	345
Section 13.3—User Access/Security Requirements	346
Section 13.4—Service Level/Performance/Capacity Requirements.....	346
Section 13.5—Data Requirements (Input, Correlative)	346
Section 13.6—Business Continuity and Recovery Requirements	346
Section 13.7—Integration/Migration Requirements	346

Section 13.8—Administrative/Backup/Archive Requirements	346
Section 13.9—Expected Life Span Requirements	346
Section 13.10—Documentation Requirements.....	346
Section 13.11—Training Requirements	346
Section 13.12—Other Nonfunctional Requirements	346
Section Fourteen: Assumptions, Dependencies, and Constraints	347
Section 14.1—Assumptions	347
Section 14.2—Dependencies	347
Section 14.3—Constraints	347
Section Fifteen: Risks and Risk Management Process	347
Section Sixteen: Solution Options.....	348
Section 16.1—Short List Solution Options.....	348
Section 16.2—Information regarding Pilot.....	348
Section 16.3—Rejected Solution Options	348
Section Seventeen: Change Management Process.....	348
Section Eighteen: Business Requirements Document Revision Log	349
Section Nineteen: Appendices.....	349
Section Twenty: Approval	349
B2. Business Requirements Document (BRD) Simple	351
Use Case Template	352
Use Case #	352

Section Zero: Positioning of the Business Requirements Document

*Goal: Common Understanding through Structured Business Analysis
and a Standard Business Requirements Document*

The business requirements document is a major deliverable representing the achievement of the business analysis milestone in a typical project management methodology. As such, it requires formal review and sign-off by the client acceptor (representing the interests of business area stakeholders). Under normal circumstances, the business requirements document is created by the senior business analyst delegated to a project.

This business requirements document template conforms to industry best practices in business analysis, and is the primary tool for structuring requirements-gathering activities. Interim feedback loops and approvals for business requirements document sections are achieved in an iterative manner, as requirements become clear over successive meetings with project stakeholders, primary and secondary users. This facilitates the final review and approval of the overall document, which by then will contain *no surprises*.

Word of Caution about Removing/Adding Sections

Do not arbitrarily add or remove sections within your business requirements document. To do so raises the risk of diluting the standard, as future teams may look to your documents for guidance in building their own reports. That being said, please use the following guidelines.

Adding Sections: Although all project business requirements documents begin with the standard template sections, the unique nature of each project may require additional sections and information. These are organized as Appendices.

Removing Sections: Because the business requirements document template has been designed as a comprehensive study, removal of specific sections of the standard template is not recommended. Doing so represents requirements detail that will not be covered, and therefore can create project risk. Whoever authorizes removal of standard sections is accountable for ownership of that risk, and any consequences that emerge as a result. This is a key point that must be understood by all members of the project team. Document the sections that have been removed, and under whose direction, within the *Risk* section of the business requirements document.

Final Note: Balance brevity with completeness, quality with quantity. You cannot document everything down to the tiniest detail and thereby eliminate all project risk. The project sponsor needs to have sufficient understanding to create an acceptable level of risk in proceeding. This will vary based on project importance and urgency, as well as the business environment within which your project lives.

Different Types of Requirements

Functional requirements can only be derived following elicitation and documentation of business and user requirements. The distinctions between these different requirements levels are important.

1. **Business Requirements:** Place the business at the center of focus, and tie the project to documented regulatory, strategic, tactical, and operational goals. If you are developing products or services for sale, customer requirements will also need to be documented. Customer requirements are covered at a high level in this section, then in detail under User Requirements.
2. **User Requirements:** Place the user at the center of focus, and describe, with flow charts, use case diagrams, use case scenarios, line of vision, and other process models, the *to-be* user experience with the new system. In some cases, especially where business processes are being modified, it may also be necessary to document the *as-is* state of user experience with the current system.

3. Functional Requirements: Place the proposed system at the center of focus, and provide a prioritized list of capabilities the system must demonstrate to satisfy business and user requirements.
4. Nonfunctional Requirements: Refer to needs that must be fulfilled related to things like the user interface, access security, availability, robustness, system failure, integration, migration, and documentation. As such, they do not deal with the actual functionality of the system, but represent key project success factors nevertheless.

Prioritizing Requirements

Ensure that your users are aware of the following interpretations regarding the prioritization of requirements:

- Must Have: Will be included in this release. These items represent core functionality and must be present. Absence of any Must Have functionality represents project failure.
- Should Have: Will be included in this release provided all Must Have requirements have been met and sufficient project resources and time remain.
- Nice to Have: Will be included in this release provided all Must Have and Should Have requirements have been met and sufficient project resources and time remain.

Section One: Glossary

This section identifies any industry or line of business jargon, acronyms, common words used in special context, and special terms that are used within the project and the business requirements document itself. Pay particular attention to acronyms that may have multiple meanings: a commonly used one and a meaning that has special significance to the business area. For example, PMO is an industry-wide acronym for Project Management Office. It may also stand for Prime Minister's Office or Preventive Maintenance Optimization. The project and business context determine the meaning.

Enter content here.

Section Two: Project Scope and Objectives Summary

This section is used to restate, in summary form, the project vision and scope statements from prior project documentation such as the Project Charter or an Opportunity Analysis. It reconfirms the understanding of project objectives, and allows for clarification of those statements that may be required due to the passage of time.

However, the business requirements document is not a project scope change device. Should scope have changed, previous project activities must be reopened, because the

reality represented by the official documentation and the signatures they contain are no longer valid.

Use this section to document high-level project deliverables (what is needed), without drilling down into details or straying into solution specifications (how we will do this).

Enter content here.

Section Three: Technology Infrastructure and Information Architecture Compliance

It is important that all project initiatives and their requirements comply with existing technology standards. Use this section to position this project within that framework. Remember that specific technologies are not normally a part of requirements. If presented as such, they often become constraints. For example, a user might state “the solution must use a Microsoft SQL Server® database.” Because this might be a violation of the approved information architecture and technology infrastructure of your company, you must verify compatibility, documenting any variance here. Note also that the only reason the existing infrastructure exists is that some prior project with a strong enough business case pushed the envelope. Your project may do the same.

Do not prejudge approval or disapproval of technology constraints that are presented as requirements. Simply raise the red flag, providing information to support a decision. However, if the decision is made during business analysis activities, document the requirement and the decision here, indicating parties involved and when the decision was made. Do not lose the history.

Enter content here.

Section Four: Intended Audience

Both readers and approvers of the business requirements document are identified here. Organizational titles and functional project roles for each individual are included. An organizational chart is very helpful in complex reader/approver environments.

Enter content here.

Section Five: Decision Making and Approval Process for the Business Requirements Document

In some cases, projects will have a number of key stakeholders who must discuss and provide interim approval for all or for specific sections of the business requirements document. However, there must always be a single client acceptor who will ultimately approve the document, representing the requirements viewpoint of the business area addressed by the project.

Within project management and business analysis, the identification of the client acceptor is a key delegation. The delegation of client acceptor may be awarded to the same individual who serves as business area project sponsor.

Pay particular attention to cultural or behavioral norms within the organization that may affect the decision-making process. This includes standard intervals for getting together (the monthly meeting), and the in-place approval and conflict resolution approach of the group.

Enter content here.

Section Six: Approach

This section describes

1. The overall project management approach for the project, including all organizations, such as service providers, system integrators and external vendors, and the roles they will play.
2. The approach that will be used for business analysis activities within this project. These may include, but are not limited to, interviews, focus groups, Requirements Joint Application Design sessions, surveys, and questionnaires. These components comprise the Requirements Work Plan.

Section 6.1—Overall Project Management Approach

Enter content here.

Section 6.2—Business Analysis Approach

Enter content here.

Section Seven: Background, Historical, and Prior Project Information

Projects exist as development or sustainment efforts within the product life cycle of systems and solution applications. Use this section to document the positioning of your project within that context. Diagrams are often helpful here.

Enter content here.

Section Eight: Business-Level Requirements: Goals, Value Proposition, and Benefits

All project initiatives exist within the organizational context of your company and consume organizational resources. As such, they must be justified by tying them to strategic, tactical, and operational goals. In some cases, there may also be regulatory

governance considerations that must be taken into account. When present, regulatory requirements are documented first. Documenting business level requirements is a critical exercise, because

1. Regulatory requirements often provide clear, nonnegotiable project constraints and quantitative success factors.
2. It assists in budgeting when there are specific moneys pre-allocated to business goals.
3. It facilitates prioritization according to the varying priorities of business goals.
4. It serves as a gauge regarding the ongoing importance of a project. That is, if the business goals or their relative priorities change during the project, your project's goals and priorities will likewise change.

Section 8.1—Regulatory Requirements

Enter content here.

Section 8.2—Related Strategic Goals (Organization Level)

Enter content here.

Section 8.3—Related Tactical Goals (Division or Department Level)

Enter content here.

Section 8.4—Related Operational Goals (Staff Level)

Enter content here.

Section Nine: User Class Profiles and Key Delegations

Users are categorized by functional groups, then by job titles as appropriate. Actual names of key individuals are supplied. The following groups are identified and described:

1. Sponsorship and stakeholders.
2. Primary Users: Those who will interact with the proposed system on a daily or regular basis, and whose job functions are directly involved with it.
3. Secondary Users: Organizations, groups, departments, and individuals who benefit from, provide input to, or derive output from the proposed system without direct involvement in its daily processes. This group also involves business or system administration personnel who must support the proposed system.

Section 9.1—Sponsors and Stakeholders

Enter content here.

Section 9.2—Primary Users

Enter content here.

Section 9.3—Secondary Users

Enter content here.

Section Ten: User and Functional Level Requirements

This section provides detailed user and functional requirements information through text and process modeling. Your company will have standard tools that have been approved for use. These may include

- Flowcharts
- Context-level data flow diagrams
- UML-based use case diagrams and scenarios
- Swim lane process work flow diagrams
- IDEF0 process models

A clear description of functional requirements defines success factors for the project. As such, functional requirements are closely tied to the project's Quality Plan. It is critical to understand that quality is conformance to documented and approved requirements and specifications. It is therefore not the same as excellence or perfection. Excellence is a moving target representing the highest level of quality achievable within the time frame of the project. Perfection is zero defects. In addition, quality has a cost expressed as follows:

$$\text{Cost of Quality} = \text{Cost of Conformance} + \text{Cost of Nonconformance}$$

Nonconformance of deliverables is frequently tied to insufficient levels of detail in the documentation of requirements. In light of this, the real focus of the business requirements document can be expressed as the establishment of project quality standards for project deliverables. An overall focus on the identification, definition, elicitation, and documentation of quantitative measures versus qualitative descriptions is a key element of functional requirement definition. You may begin your descriptions using qualitative language such as *improved* or *faster*, but follow this with actual metrics.

Enter content here.

Section Eleven: Additional Information regarding Functional Requirements Related to Output and Reporting

This section describes the requirements of primary and secondary users related to screen, transmitted, and hardcopy output from the proposed system. This includes but is not limited to

- Ad hoc queries
- Scheduled/batch reports
- Audit or control reports

Also included here are

- Recipient information maps: who gets what information (need to know, should know, wants to know)
- Business rules that govern output generation: who, what, where, when, why, how much

Enter content here.

Section Twelve: Conceptual Data Model

This is a very high-level, requirements-oriented set of diagrams that will be elaborated upon by database analyst personnel during solution design. This section may be limited to simply identifying database systems and their interactions with project applications. Alternatively, it may include actual database schema diagrams and descriptions. The amount of detail will vary by project and company policy.

Enter content here.

Section Thirteen: Nonfunctional Requirements

This section documents requirements that are not directly related to the functionality of the proposed system. Like functional requirements, the clear, quantitative definition of these requirements feeds the project's Quality Plan. Please be conscientious in your investigation of nonfunctional requirements, as these are often overlooked or given insufficient attention.

Section 13.1—Operational Environment

Enter content here.

Section 13.2—User Interface Requirements

Enter content here.

Section 13.3—User Access/Security Requirements

Enter content here.

Section 13.4—Service Level/Performance/Capacity Requirements

Enter content here.

Section 13.5—Data Requirements (Input, Correlative)

Enter content here.

Section 13.6—Business Continuity and Recovery Requirements

Enter content here.

Section 13.7—Integration/Migration Requirements

Enter content here.

Section 13.8—Administrative/Backup/Archive Requirements

Enter content here.

Section 13.9—Expected Life Span Requirements

Enter content here.

Section 13.10—Documentation Requirements

Enter content here.

Section 13.11—Training Requirements

Enter content here.

Section 13.12—Other Nonfunctional Requirements

Enter content here.

Section Fourteen: Assumptions, Dependencies, and Constraints

Assumptions: All projects operate in a less-than-perfect world. Not everything can be officially verified as existing or available ahead of time. These *unknowns* are documented in project assumptions. An example might be “The project assumes the continued availability of funding following the upcoming merger.”

Dependencies include, but are not limited to, the availability of project resources, applications, and systems that interact with this one, hardware, facilities, equipment, business processes, and regulatory approvals. Of particular importance is the dependency on the availability of project stakeholders and users, and conformance to approval and change management processes.

Constraints are those regulatory, technological, or business realities that legitimately constrain solution development. An example might be “The new system must be built in Oracle™.” Although this example might sound, on the surface, like a specification (and therefore not part of a business requirements document), it becomes a constraining requirement when stated up front. It is for this reason that users must be cautioned against careless statement of constraints.

In essence, this section allows you to document that which cannot be ascertained in advance. These items may feed the *Risks and Risk Management* section, which follows.

Section 14.1—Assumptions

Enter content here.

Section 14.2—Dependencies

Enter content here.

Section 14.3—Constraints

Enter content here.

Section Fifteen: Risks and Risk Management Process

This section documents project risks

1. That have been uncovered as a result of business analysis activities
2. Related to the business analysis itself (such as not having adequate time to perform required analysis and nonavailability of key requirements voices)

In addition, this section documents

1. Risk mapping: likelihood versus impact potential
2. Who owns risks, and deadlines for risk resolution
3. Contingency plans

Enter content here.

Section Sixteen: Solution Options

As a business analysis vehicle, the business requirements document focuses on requirements, not specifications or solutions. Nevertheless, as requirements are elicited and documented, discussion around solution options will occur. This section is used to document those options that have been considered to date, rejected, or approved for further investigation. It is important to include rejected entries for the historical record of the project and to preempt future readers who may ask “did they consider such-and-such?”

Remember that solution options must be derived based on clear understanding of project requirements, assumptions, dependencies, and constraints.

Section 16.1—Short List Solution Options

Enter content here.

Section 16.2—Information regarding Pilot

Note: This section is included only if a pilot has, in fact, been proposed. It will cover the specific target user groups, subset of requirements to be addressed, funding, timing, and key personnel required. Do not use this section as a project plan for a pilot, although it will identify the pilot’s key personnel.

Enter content here.

Section 16.3—Rejected Solution Options

Enter content here.

Section Seventeen: Change Management Process

Because business analysis activities are exploratory and iterative up to the approval of the business requirements document, it is likely that some requirements will evolve throughout the process. This consumes project resources and so must be governed by change management. For example, a requirement that takes one day to elicit,

document, and approve, but that changes 30 times during business analysis, will consume 30 project days worth of appropriate resources.

This section provides information, normally documented in official policy, regarding the change management process to be used for this project. It also includes details regarding any additional change management that must be applied due to the special needs or unique nature of this project.

If there is an official change management policy that is documented elsewhere (such as in the Project Plan), you may simply reference it here.

Enter content here.

Section Eighteen: Business Requirements Document Revision Log

This section documents requirements that have changed over the course of successive documentation iterations during business analysis activities. Pay particular attention to requirements with ongoing adjustment. These are high-risk areas that may represent

1. Lack of clear business process definition
2. Unclear reporting requirements
3. Areas of regulatory flux
4. Unclear secondary user hand-off points
5. Unclear governance related to specific requirements

If you find that a majority of Must Have requirements cannot be nailed down, you may have a situation where the project itself must be reassessed at a scope level.

Enter content here.

Section Nineteen: Appendices

Each Appendix must have

- A separate header, numbered A through Z, with an appropriate descriptive title. For example: “[Appendix A](#)—Regulatory Requirements.”
- Note: Use the Heading 1 Style for each Appendix header. This style will automatically insert a page break.
- A lead in paragraph that states the importance of the data to this report.
- A closure, centered on a separate line, that repeats the header, such as “End of [Appendix A](#)—Regulatory Requirements.”

Enter content here.

Section Twenty: Approval

This section documents the approvals required for sign off of the business requirements document and establishment of the Requirements Analysis milestone. Each signatory

is described by both organizational title and project functional role. Required signatories are the business analyst and the client acceptor. Other signatories may be required by your company policy.

This document has been approved as the official Business Requirements Document for the [name of project] project, and accurately reflects the current understanding of business requirements. Following approval of this document, requirements changes will be governed by the project's change management process, including impact analysis, appropriate reviews, and approvals, under the general control of the Project Plan and according to company policy.

Prepared by

Business Analyst

Date

Approved by

Client Acceptor

Date

All products, services, and company names used within this template are trademarks or registered trademarks of their respective owners.

B2. Business Requirements Document (BRD) SimpleBusiness Requirements Document
XYZ Product

1. As-Is Scenario
 - a. Background
 - b. Context Diagram
 - c. Business Processes
 - d. Business Rules
 - e. Business Objectives
 - f. Business Constraints
 - g. Regulatory Constraints
 - h. Process Models
 - i. Data Models
 - j. Data Flow Diagrams
2. Product Vision and Scope
 - a. Vision Statement
 - b. Key Features
 - c. Exclusions
 - d. Impacted Stakeholders
 - e. Context Diagram
 - f. Assumptions
 - g. Constraints
3. Business Requirements
 - a. Business Processes
 - b. Business Rules
 - c. Business Objectives
 - d. Business Constraints
 - e. Regulatory Constraints
4. User Requirements
 - a. Use Case 1
5. Nonfunctional Requirements
 - a. Performance
 - b. Safety
 - c. Security
 - d. Quality Attributes
 - e. User Documentation
6. Data Requirements
 - a. Entity Relationship Diagram
 - b. Data Flow Diagram

Appendix A: Glossary

Appendix B: Issues Lists

*Use Case Template**Use Case #***General Characteristics**

Intent	[a summary statement of the purpose of the use case]
Level	[one of business level, system level, or component]
Author	[name of use case author(s)]
Last update	[date last updated/change history]
Status	[one of incomplete, under review, finalized, etc.]
Primary actor	[role name for the primary actor, <optional description>]
Secondary actors	[role names of other actors (could be systems) relied upon to accomplish use case]
Preconditions	[what we expect is already the state of the world]
Assumptions	[any assumptions relative to this use case]
Trigger	[the event that starts the use case]
Success postcondition	[the state of the world upon successful completion]
Failed postcondition	[the state of the world if use case abandoned]
Overview	[description in words that encompasses all scenarios]

Main Success Scenario

STEP	ACTION
S	[description in words of the main success scenario]
1	["This use case starts when ..." followed by the trigger.]
2	[step description... <"included" use case pointer>]
3	[step description... <"included" use case pointer>]
4	["This use case ends when ..." the final step in main success scenario.]

Extension Scenarios

STEP	BRANCHING ACTION
1	[description in words of the extension scenario] [any scenario other than the main scenario. Could be valid but less common path or error path]

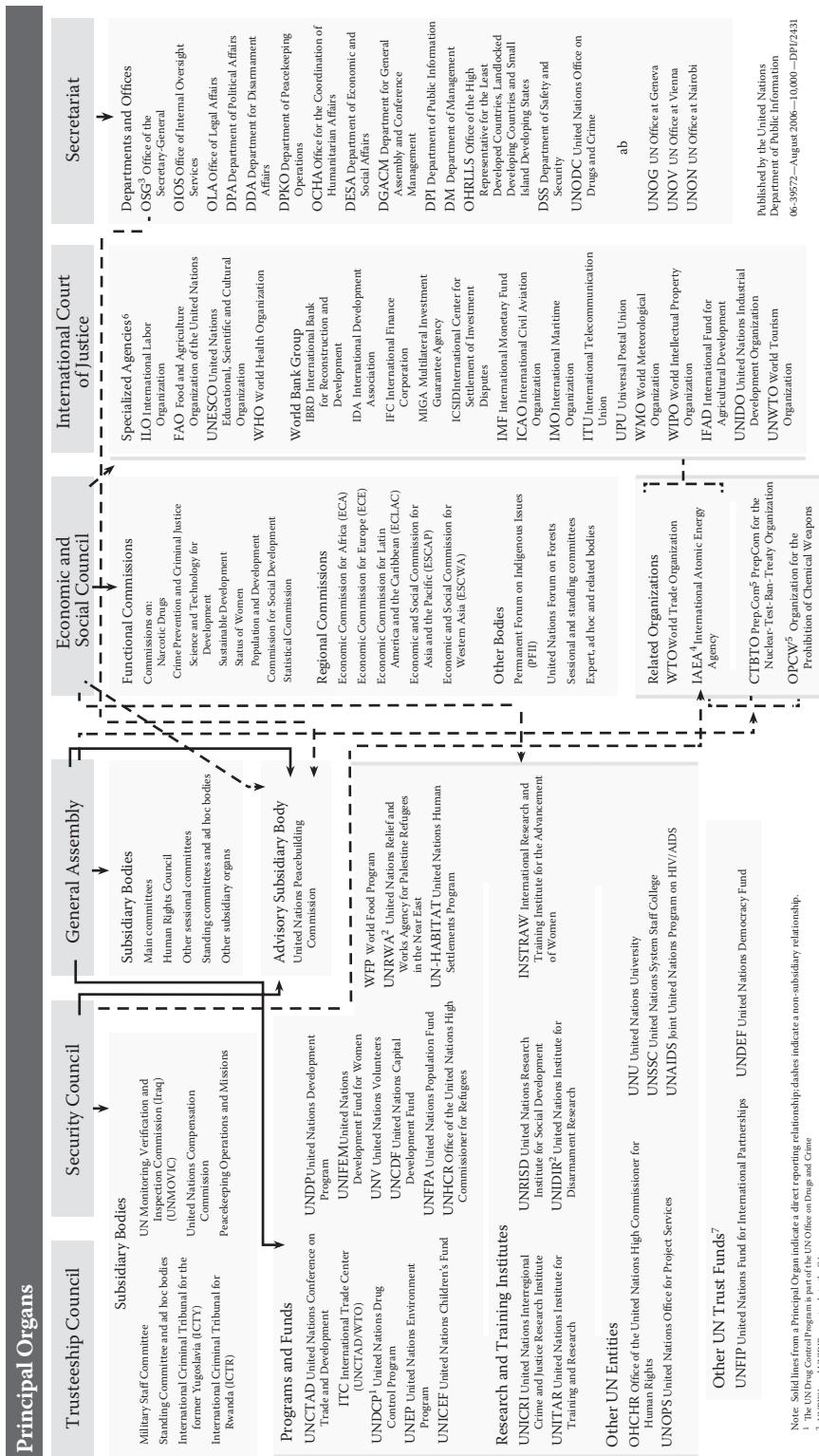
Related Information

Performance	[process and/or system performance]
Frequency	[how often it is expected to happen]
Open issues	[list of issues awaiting decision]
Future considerations	[list of all requirements or possible requirements that have been deferred to future increments]
Comments	

Appendix C: United Nations Organizational Chart

The

Principal Organs



Index

Page numbers followed by f and t indicate figures and tables, respectively.

A

- Acceptance criteria, defined, 331
- Acceptance test, 229
- Accounts payable, Swede-Mart (case study), 236
- Activity(ies)
 - business analysis, 60–61, 60f
 - diagrams, 194–196, 195f, 318
 - solutions, for Swede-Mart (case study), 291–299
- Actors, in use cases, 310
 - primary, 192, 323
 - secondary, 192, 328
- Affinity diagramming, 119–120, 120f
- Agenda, 311, 313
- Agile approaches, 43–49
 - DSDM, 46–48, 47f
 - JIT requirements gathering, 48–49
 - overview, 43–44
 - RAD, 44–45, 44f
 - Scrum™, 45–46
- Agile development
 - defined, 36
 - perspective on business analysis, 27
- Alternative scenarios, use case, 191
- Amazon.com, 112
- Ambler, Scott, 48
- Analogous estimates, 68
- Analogous technique, 323
- Analysis
 - business, *see* Business analysis
 - business rules, 330
 - CBAP exam, *see* Certified Business Analysis Professional (CBAP) exam
 - decision, 308
 - document, 311, 323
 - force field, 329, 331
 - IIBA, *see* International Institute of Business Analysis (IIBA)
 - impact, defined, 28
 - interface, 70, 311, 328
 - MoSCoW, 138
 - potential value and recommend solution, 218–219
 - requirements, design definition and, *see* Requirements analysis and design definition
 - risks, network diagram, 69, 69f
 - root-cause, 28, 307, 319, 324
 - stakeholder, 38, 293
 - strategy, *see* Strategy analysis
 - structured analysis techniques, 28
 - SWOT, 145–146, 146f, 150, 151, 152, 318–319, 322
 - tools, cost, 70

- Analyst time, cost, 70
- Analytical hierarchy process (AHP), 126
- Analytical thinking, 28
- Anonymous brainstorming, 119, 312
- Applicability**
- business rules, 191–192
 - SEI-CMMI and, 19–21
 - requirements development, 21
 - requirements management, 20–21
- Approach to finding solution, determining, 152
- Approvals
- process, for BRD, 56, 341–342, 349–350
 - requirements, 138–139, 317
 - from stakeholders, 39
- Architecture, requirements, 202–214
- categorization, 204–205, 204f
 - combining structures, 214
 - design options, defining, 214–218, 216f, 217f, 218f
 - detail, need, 205–207
 - potential value and recommend solution, analysis, 218–219
 - purpose-based classification, 213–214
 - regulations, industry standards, and corporate policies, 211–213
 - sequence-oriented classification, 208–213
 - stakeholder-based classification, 207–208
 - taxonomy, 202–205, 203f
- As-is documentation, 147, 149, 315, 331
- Assessment**
- capability gaps, 150–152, 322
 - cultural, 331
 - requirements changes, 138
 - risks, 66–67, 66f
 - initial, conducting, 157–158
 - technical, 332
 - vendor, 312
- Assumptions**
- BRD template, 347
 - for planning purposes, 153–154
 - for requirements, 210–211
 - use cases and, 192
- Asynchronous flow, in sequence diagram, 327
- Attributes
- documentation, 179–180, 179f
 - identifying, 178
 - requirements, 309
- Auditability, 39
- Author, use cases and, 192
- B**
- BABOK®*, see *Business Analysis Body of Knowledge® (BABOK®) Guide*
- Back-up and recovery, use cases and, 192
- Behavioral characteristics, business analyst's skills, 29
- Behavior diagrams, 185, 186–194, 188f, 190f
- Benchmarking, 142, 215, 311, 322
- Benefits, costs *vs.*, 156–157
- Best practices
- ERD, 181
 - evaluation, 112–113
 - FDD, 176
 - focus groups, 111
 - idea-generating techniques (brainstorming), 121
 - interviews, 87
 - JAD, 110
 - market research, 112
 - observation, 89
 - prototyping, 115
 - requirements prioritization techniques, 126
 - storyboarding, 117
 - studying
 - existing systems, 90
 - interfaces, 92
 - surveys, 94
- Beta/pilot testing, 229
- Blogs, 122
- Body of knowledge, defined, 14–15
- Boehm, Barry, 41
- Booch, Grady, 185
- Bottom-up estimates, 68
- Brainstorming, 117–121
- advantages and disadvantages, 121
 - affinity diagramming, 119–120, 120f
 - anonymous, 119, 312
 - basic, 117–118
 - best practices, 121

- mind mapping, 306
with one person, 120–121
problem, 311
purpose of, 117
for virtual team, 121
- BRD, *see* Business requirements document (BRD)
- Business
architecture, perspective on business analysis, 27
goals and objectives, 153, 320
intelligence, perspective on business analysis, 27
knowledge, 29
location model, 162, 163f
models, 160–166
entity, 164–165, 164f, 321
events, 162, 163–164, 321, 325
infrastructure, 161–162, 161f, 321
location, 162, 163f
organization chart, 160, 354
process, 165–166, 165f
need, 320, 321
defining, 145–150, 145f, 146f, 148f, 149f
process models, 165–166, 165f
requirements, 339
study, DSDM project cycle, 47
understanding, 159–160, 303
- Business analysis
body of knowledge in, 11–12
defined, 6, 17
performance measurement for, 309
planning, 58–63; *see also* Planning and monitoring
approach, 34–49
governance, 56–57, 307
information management, 57
knowledge area, 306
manage changes to requirements, 63
overview and background, 59
performance improvements, 57–58
project activities, 60–61, 60f
requirements risk plan, 62–63
resource plan, 62
roles and responsibilities, 61, 61f, 309
scope and deliverables, 59–60
solution evaluation, 18
strategy analysis, *see* Strategy analysis
- Business Analysis Body of Knowledge® (BABOK®) Guide*
BABOK v3, 6, 8, 17, 18, 306
BRD, reviewing, 128–129
business analyst, role, 6, 7–8, 7f, 61, 61f, 142–143
business analyst's skills, 28–31
analytical thinking and problem solving, 28, 303
behavioral characteristics, 29
business knowledge, 29
communication skills, 29–30
interaction skills, 30–31
tools and technology, 31
deliverables, 317
early days of computer systems, 3–5, 4f
IIBA, *see* International Institute of Business Analysis (IIBA)
key concepts, 25–26, 26f
MoSCoW analysis, 138
overview, 1–10, 25–31
perspectives, 26–28
agile, 27
architecture, business, 27
information technology, 27
intelligence, business, 27
process management, business, 28
- PMI, 5, 8
purposes, 2
for requirements gathering, 11
strategy analysis, defined, 141
task, 306
- Business Analysis Core Concept Module™ (BACCM™)
defined, 25, 306
key concepts, 25–26, 26f, 306
- Business analysts
boundaries, understanding, 85–86, 85f
communication skills, 149–150
conflict resolution for, 31
domain subject-matter expert, 305
future, envisioning, 86
from IT side, 29
job description for job posting, 304
Maslow's hierarchy of needs for, 305
observation/job shadowing, 87–89
participant, in JAD sessions, 98
problem, defining, 86

- project manager, 17, 306
 requirements of solution, determining, 86
 responsibilities of, 6, 7–8, 7f, 60, 61, 61f, 142–143, 221, 305
 review requirements, 315
 risk management, 62–63
 role, 6, 7–8, 7f, 61, 61f, 142–143, 303
 role of facilitator, 305
 root cause, identifying, 86
 skills, 28–31
 - analytical thinking and problem solving, 28, 303
 - behavioral characteristics, 29
 - business knowledge, 29
 - communication skills, 29–30
 - interaction skills, 30–31
 - tools and technology, 31
 sunk cost, 305
 tasks in *BABOK*, 305
 in testing process, 330
 Business case, 156–157, 319, 320, 321, 322, 323
 Business requirements document (BRD)
 - approval, 56
 - as-is section, 147, 149, 315, 331
 - business analysis
 - governance of, 56
 - plan, deliverables, 59
 - content and level of detail in, 31
 - creating, 69, 70, 78
 - business analyst for, 78
 - defined, 31, 39
 - reviewing, 128–129, 135–136
 - stakeholders for, 126–127, 127t
 - templates, 147, 335–352
 - appendices, 349
 - approach, 342
 - approvals, 349–350
 - assumptions, dependencies, and constraints, 347
 - background, historical, and prior project information, 342
 - business-level requirements, goals, value proposition, and benefits, 342–343
 - change management process, 348–349
 - comprehensive, 336–350
 - decision making and approval process, 341–342

 glossary, 340
 information regarding functional requirements, 345
 intended audience, 341
 positioning of, 338–340
 removing/adding sections, 339
 requirements, types, 339–340
 revision log, 349
 risks and risk management process, 347–348
 scope and objectives summary, 340–341
 simple, 351–352
 solution options, 348
 structured business analysis and standard BRD, 338
 technology infrastructure and information architecture compliance, 341
 user and functional level requirements, 344
 user class profiles and key delegations, 343–344
 Business rules, 191–192, 321, 323, 326, 330
- ## C
- Capability gaps, assessment, 150–152, 322
 Capability Maturity Model® Integrated (CMMI) certification
 - SEI-CMMI, 12, 37
 - applicability and, 19–21
 - requirements development, 21
 - requirements management, 20–21
 Cardinality
 - defined, 178
 - maximum, 179
 - minimum, 178
 - symbols, 179
 CASE (computer aided systems engineering) tools, 44, 45, 46, 95
 Cash flow analysis, 323
 Categorization, requirements, 204–205, 306
 Certifications
 - CMMI, *see* Capability Maturity Model® Integrated (CMMI) certification
 - demand for, 12
 - ISO, 12

- Certified Business Analysis Professional (CBAP) exam
early days of computer systems, 3–5, 4f
IIBA, *see* International Institute of Business Analysis (IIBA)
overview, 1–10
PMI, 5, 8
purposes, 2
- Change control board, 307
- Change-driven approach, IIBA, 37, 308, 309
- Change management process, 348–349
- Characteristics, JAD project, 95
- Charters, project, 92, 153, 156, 296, 318, 340
- Chen, Peter, Dr., 177
- Clarifier, facilitator in JAD sessions, 97
- Class diagrams, 185, 186f, 326
- Classification
purpose-based classification, 213–214
sequence-oriented, 208–213
stakeholder-based, 207–208
- Class interfaces, 185, 186f
- Closed-ended questions, 83, 84, 318
- CMMI, *see* Capability Maturity Model® Integrated) certification
- Code and fix, methodology, 36
- Collaboration tools, 122
- Combining structures, 214
- Commercial off-the-shelf (COTS) software, 157
- Communication(s), 327
developers and customers, 5
formal, 309, 311
frequent, short, informal, 308
issues, language, 78
management, 16
message, 304
plan, 312
requirements, 131–132, 132f, 318
skills, 29–30
stakeholder, 55–56, 56f
- Computer aided systems engineering (CASE)
tools, 44, 45, 46, 95
- Computer systems, early days of, 3–5, 4f
- Conceptual data models, 180, 320, 324, 327, 345
- Conflict resolution, for business analysts, 31
- Constraints, 322, 327
BRD template, 347
for requirements, 210–211
- solution scope, defining, 154
use cases and, 192
- Context diagram, 154–156, 155f, 311, 318
- Controllability
defined, 66
dimension, 66–67
- Controlling, risks, 67
- Conversion process, 230
- Coordinator, facilitator in JAD sessions, 97
- Corporate policies, requirements architecture, 211–213
- Cost(s)
benefits *vs.*, 156–157
management, 16
nonfinancial, 156, 332
opportunity, 222, 329
requirements plan, 69–70
sunk, 305
virtual environments, tools, 122
- Counselor, facilitator in JAD sessions, 97
- Coverage matrix, 316
- Crashing, 69
- Create–read–update–delete (CRUD) matrix, 197, 198f, 325
- Creativity lab, defined, 119
- Cultures
assessment, 331
concern with global team, 78–79
differences, 54–55
- Customer(s)
business requirements package, 127, 127t
interviews, 79–87
advantages and disadvantages, 87
boundaries, understanding, 85–86, 85f
future, envisioning, 86
high-level need for, 80–81
homework, 81
initial list of questions, drafting, 81–82, 82f
open-ended, 80
problem, defining, 86
questions, types of, 82–84, 82f
requirements of solution, determining, 86
root cause, identifying, 86
sample interview agenda, 84f
schedule, 81
matching solution to needs of, 222–223
- participant, in JAD sessions, 99

- selecting elicitation technique, 77
 unrealistic expectations, 311
 Customer satisfaction, evaluating, 229–230
 C.V. Green, 9, 42, 142, 144, 146, 149, 149f, 161, 162, 163f, 164, 165, 212
- D**
- Data flow diagrams (DFDs), 166, 181–184, 181f, 182f, 183f, 184f, 323, 327, 328, 332
 Data mining, 307
 Data models, 313, 328, 329
 conceptual, 180, 320, 324, 327, 345
 IIBA tasks, requirements analysis, 177–181, 177f, 178f, 179f
 logical, 180
 physical, 180
 to-be, 86, 127, 230, 331
 Decision analysis, 308
 Decision making, 317, 323, 341–342
 Decision package, preparing, 158–159, 159f
 Decision tree, 312
 Deliverables
 BABOK, 317
 business analysis, planning, 59–60
 nonconformance, 344
 SDLC, 13, 13f
 Density dotting, 125–126, 314
 Dependencies, 69, 347
 Design and build, DSDM project cycle, 47
 Design options, defining, 214–218, 216f, 217f, 218f
 Detailed estimates, 68
 Developers, in JAD sessions, 100
 Development
 process, 330
 team, business requirements package, 127, 127t
 tools, cost, 70
 DFDs, *see* Data flow diagrams (DFDs)
 Diagrams
 activity, 194–196, 195f, 318
 affinity, 119–120, 120f
 behavior, 185, 186–194, 188f, 190f
 class, 185, 186f, 326
 context, 154–156, 155f, 311, 318
 DFDs, *see* Data flow diagrams (DFDs)
- ERD, 173, 177, 177f, 179, 180–181, 185, 326, 327
 FDD, *see* Functional decomposition diagram (FDD)
 interaction, 197
 level 0 data flow, 154–156, 155f
 network, analyzing risks, 69, 69f
 onion, 307, 310
 requirements development process, 318
 sequence, 184, 197, 326, 327
 state, 196, 197f, 326, 327, 329
 structure, of UML model, 185, 186f
 swim lane, 111, 165, 194, 326, 344
 text document and, 323
 use cases, example of, 188–189, 188f
 Diplomat, facilitator in JAD sessions, 97
 Disagreement, reason for, 109
 Discovery sessions, 94–110; *see also* Joint Application Development (JAD) sessions
 Distribution center, Swede-Mart (case study), 234–235
 Documentation
 analysis, 311, 323
 as-is section, 147, 149, 315, 331
 assumptions, 154, 304
 BRD, *see* Business requirements document (BRD)
 business analysis, scope and deliverables, 59–60
 for entity, attribute, and relationship, 179–180, 179f
 existing system, 90
 expectations, 304
 interfaces, 69
 matrix, 197–198, 198f
 problem, 320, 322
 requirements, sign-off of, 316
 stakeholder categorization, 50, 50f
 text, 198–200, 329
 diagrams and, 323
 use cases, *see* Documentation
 waterfall requirements, 39, 39f
 Dollar prioritization, 123–124, 123f
 Domain subject-matter expert, 305
 Duplication, of information, 59
 Dynamic systems development method (DSDM), 44, 46–48, 47f

E

Early days, of computer systems, 3–5, 4f
Elicitation and collaboration, 73–133
 activity solutions, for Swede-Mart (case study), 295
 best practices, evaluation, 112–113
 communicate requirements, 131–132, 132f
 customer interviews, 79–87
 advantages and disadvantages, 87
 boundaries, understanding, 85–86, 85f
 future, envisioning, 86
 high-level need for, 80–81
 homework, 81
 initial list of questions, drafting, 81–82, 82f
 open-ended, 80
 problem, defining, 86
 questions, types of, 82–84, 82f
 requirements of solution, determining, 86
 root cause, identifying, 86
 sample interview agenda, 84f
 schedule, 81
discovery/JAD/facilitated sessions, 94–110
 business analyst, 98
 characteristics, 95
 creation, 95–96
 developers, 100
 facilitation, 107–109
 facilitators, 96–98, 104–105
 history, 95
 JAD processes, 101–107, 101f
 observers, 101
 overview, 94
 participants, 96–101
 scribe, 98–99
 sponsors, 100
 subject-matter experts, 99–100
 user (customer), 99
focus groups, 110–111
idea-generating techniques
 (brainstorming), 117–121
 advantages and disadvantages, 121
 affinity diagramming, 119–120, 120f
 anonymous, 119
 basic, 117–118
 best practices, 121

with one person, 120–121
for virtual team, 121
IIBA view, 74, 75–76
 activity, conducting, 75–76
 business analysis information, communicating, 76
 preparation, 75
 results, confirming, 76
 stakeholder collaboration, managing, 76
market research, 111–112
observation/job shadowing, 87–89
overview, 74
prototyping, 113–115
requirements, iterative, 315, 324
requirements package, preparation, 126–131
 customers, 127, 127t
 development team, 127, 127t
 executives, 127t, 128
 external organizations, 127t, 128
 internal organizations, 127t, 128
 modeling techniques, 128
 prototyping, 128
 text, 128
 users, 127, 127t
requirements prioritization techniques, 122–126
 advantages and disadvantages, 126
 AHP, 126
 best practices, 126
 density dotting, 125–126
 dollar approach, 123–124, 123f
 forced-pair ranking, 124–125, 124f, 125f
 risks, 307
 storyboarding, 115–117
 studying
 existing systems, 89–90
 interfaces, 90–92
 surveys, 92–94
technique for gathering requirements, selection, 76–79, 315
 for customers, 77
 for different categories, 77
 globalization, impact of, 78–79
technique of interface analysis, 310
virtual environments, tools for, 122

- E-mail, 122
 Enterprise architecture, 308
 Enterprise requirements, 304
 Enterprise resource planning (ERP) system, 330
Entities
 documentation, 179–180, 179f
 identifying, 178
 Entity life cycle diagram, 327
 Entity models, business, 164–165, 164f, 321
 Entity relationship diagram (ERD), 173, 177, 177f, 179, 180–181, 185, 326, 327
 Estimates, types, 68
Evaluation
 best practices, 112–113
 customer satisfaction, 229–230
 potential consolidation of use cases, 187
 solution, *see Solution evaluation*
 Event models, business, 162, 163–164, 321, 325
 Evolutionary prototype, 114, 329
 Exceptions, use cases, 191
 Executives, business requirements package, 127t, 128
 Existing systems, studying, 89–90
 Experienced users, 53–54
 Extends use case, 188, 189, 325
 External users, 54
 eXtreme Programming (XP), 49
- F**
- Face-to-face meeting, 77, 314
 Facilitated sessions, 107–110; *see also Joint Application Development (JAD) sessions*
 Facilitation, for JAD sessions, 107–109
 fears, 107–108
 Facilitators
 in JAD sessions, 96–98, 104–105
 neutral and experienced, 305, 312
 Fast tracking, 69
 FDD, *see Functional decomposition diagram (FDD)*
 Feasibility studies, 320, 321
 capability gaps, assessment, 150–152
 DSDM project cycle, 47
 technical, 314
- File sharing, 122
 Fishbone diagram, 319
 Flowcharts, 306
 Focus groups, 110–111, 310, 314, 323, 331
 Follow up, JAD processes, 107
 Forced-pair ranking, 124–125, 124f, 125f, 216, 216f, 312
 Force field analysis, 329, 331
 Forcing, 304
 Forward traceability, 317
 Full-time users, 54
 Functional decomposition diagram (FDD), 325, 326, 328, 332
 best practices, 176
 boundaries of interview, 85–86, 85f
 creating, 174–176, 174f
 for documenting requirements, 173–176, 174f, 175f
 hierarchical model, example, 166
 level of, 181–182, 181f
 process description, 175–176, 175f
 strengths and weaknesses, 176
 Functional model iteration, DSDM project cycle, 47
 Functional requirements, 209, 305, 318, 339, 340, 345
- G**
- Gap analysis, 128, 218, 268, 269, 321
 General Motors (GM), 4
 Geographical differences, 55
 Globalization, impact of, 78–79
 Goals
 business, 153, 320
 JAD processes, 101
 for requirements management, 20
 Governance, of business analysis, 56–57, 307
 Ground rules, 310, 315
- H**
- Hawthorne effect, 88, 89
 Heterogeneous focus groups, 110–111
 Homogeneous focus groups, 110
 Horizontal prototype, 114
 Human resource management, 16

I

IBM, 95

Idea-generating techniques (brainstorming), 117–121

- advantages and disadvantages, 121
- affinity diagramming, 119–120, 120f
- anonymous, 119
- basic, 117–118
- best practices, 121
- with one person, 120–121
- for virtual team, 121

Identifier (ID), use case, 189

Impact analysis, defined, 28

Impacted organizations, 154–156, 155f

Implementation

- approach, example, 321
- DSDM project cycle, 47
- requirement, 304, 325
- solution, 230–231

Includes use case, 188, 189, 191

Industry background, Swede-Mart (case study), 234

Industry standards, requirements architecture, 211–213

Information

- availability, 313
- management, business analysis, 57

Information technology (IT)

- business analysts from, 29
- perspective on business analysis, 27

Infrastructure models, 161–162, 161f, 321

Initial risk assessment, conducting, 157–158

Input and task, relationship, 306

Input–process–output (IPO) models, 166

Integrated CASE (I-CASE) tools approach, 46

Integrated project management, defined, 37

Integration management, 15

Integration test, 227–228

Interaction diagrams, 197

Interaction skills, 30–31

Interfaces

- analysis, 311, 328
- studying, 90–92
- types, 210

Internal users, 54

International Institute of Business Analysis (IIBA)

BABOK, see *Business Analysis Body of Knowledge® (BABOK®) Guide*

BACCM, 25, 306

body of knowledge for, 15

classification system, 169

framework and standards, overview, 17–19

framework for business analysis, 6, 8

goal, 6

requirement, definition of, 303

for requirements gathering, 11

six knowledge areas, 18

tasks, requirements analysis and design definition, 172–179

architecture, defining, 202–214

categorization, 204–205, 204f

combining structures, 214

conceptual data models, 180

data models, 177–181, 177f, 178f, 179f

design options, defining, 214–218, 216f, 217f, 218f

detail, need, 205–207

DFD, 181–184, 181f, 182f, 183f, 184f

logical data models, 180

matrix documentation, 197–198, 198f

physical data models, 180

potential value and recommend solution, analysis, 218–219

process models, 173, 174–176, 174f, 176f

purpose-based classification, 213–214

regulations, industry standards, and corporate policies, 211–213

selection of model, 198

sequence-oriented classification, 208–213

specify and model requirements, 173–200, 173t

stakeholder-based classification, 207–208

taxonomy, 202–205, 203f

text documentation, 198–200

UML family, 184–197; *see also Unified Modeling Language (UML) models*

validation, 202, 328

verification, 201–202, 326, 328

- tasks, requirements life cycle management, 136–139
 approval, 138–139
 changes, assessment, 138
 maintenance, 137
 prioritization, 137–138
 traceability, 136–137, 136f
- tasks, solution evaluation, 222
- tasks, strategy analysis, 143–157, 321
 approach to finding solution,
 determining, 152
 assess capability gaps, 150–152
 assumptions, 153–154
 business case, defining, 156–157
 business goals and objectives, 153
 business need, defining, 145–150, 145f,
 146f, 148f, 149f
 constraints, 154
 impacted organizations, 154–156, 155f
 overview, 143–145
 scope statement, 154
 solution scope, defining, 152–156
- view
 elicitation and collaboration, 74, 75–76
 planning and monitoring, 34–58; *see also* Planning and monitoring
- International Organization for Standardization (ISO) certifications, 12
- Interviews
 advantages and disadvantages, 87
 boundaries, 85–86, 85f, 313
 customer, 79–87
 face-to-face, 77, 314
 future, envisioning, 86
 high-level need for, 80–81
 homework, 81
 initial list of questions, drafting, 81–82,
 82f
 preparing, 312
 problem, defining, 86
 questions, types of
 closed-ended, 83, 84, 318
 open-ended, 80, 82–83, 84, 311
 probing, 83, 84
 sample, 82f
 validating, 83, 84
- requirements of solution, determining, 86
 root cause, identifying, 86
- sample interview agenda, 84f
 schedule, 81
 with store buyer, Swede-Mart (case study), 238
 structured, 311
 waterfall approach, 40
- Inventory operations, Swede-Mart (case study), 234–235
- Ishikawa diagram, 322
- Iterative life cycle, 41–43, 41f, 308, 314
- J**
- Jacobson, Ivar, 185, 186
- Job shadowing, 87–89, 315
- Joint Application Development (JAD)
 sessions, 94–110
 advantages and disadvantages, 110
 characteristics, 95
 creation, 95–96
 facilitation, 107–109
 fears, 107–108
 history, 95
 overview, 91, 94
 participants, 96–101
 business analyst, 98
 developers, 100
 facilitators, 96–98, 104–105
 observers, 101
 scribe, 98–99
 sponsors, 100
 subject-matter experts, 99–100
 user (customer), 99
 processes, 101–107, 101f
 conducting, 103–107, 104f
 follow up, 107
 goals and objectives, 101
 preparation, 101, 102–103, 102f,
 103f
 purposes of, 105
- Joint Requirements Planning, 95
- Just-in-time (JIT) requirements gathering, 48–49
- K**
- Kickoff meeting, 71, 103, 308
- Knowledge, business, 29

L

- Language concerns, 54–55, 79, 93
- Laying out, tasks, 68–69, 69f
- Level 0 data flow diagram, 154–156, 155f
- Life cycle
 - definitions, 12–14, 13f, 14f
 - iterative, 41–43, 41f, 308, 314
 - management, requirements, 135–139
 - approval, 138–139
 - changes, assessment, 138
 - IIBA tasks, 136–139
 - maintenance, 137
 - overview, 135–136
 - prioritization, 137–138
 - traceability, 136–137, 136f
- SDLC
 - defined, 12
 - for IT projects, 13
 - phases, deliverables, and roles and responsibilities, 13, 13f
 - project management processes, 13, 14, 14f
 - waterfall, 36–37, 38–41, 39f
- Logical data models, 180

M

- Main scenario, use case, 191
- Maintenance, requirements, 137
- Management
 - communication, 16
 - cost, 16
 - human resource, 16
 - information, 57
 - integrated project, 37
 - integration, 15
 - PMBOK™ Guide, see Project Management Body of Knowledge, (PMBOK™) Guide*
 - PMI, *see Project Management Institute (PMI)*
 - procurement, 16
 - project management processes, SDLC, 13, 14, 14f
 - quality, 16
 - requirements, 20–21
 - risk(s), 16
 - approach, development, 64
 - scope, 16
- stakeholder, 16
- time, 16
- Total Quality, 22
- Market research, 111–112
- Martin, James, 44, 95, 113
- Matrix documentation, 197–198, 198f
- Maximum cardinality, defined, 179
- Mediator, facilitator in JAD sessions, 97
- Merriam-Webster's Collegiate Dictionary*, 202
- Message, 304
- Mind mapping, 306
- Minimum cardinality, defined, 178
- Model(s)
 - best description of, 329
 - business, 160–166
 - entity, 164–165, 164f, 321
 - events, 162, 163–164, 321, 325
 - infrastructure, 161–162, 161f, 321
 - location, 162, 163f
 - organization chart, 160, 354
 - process, 165–166, 165f
 - conceptual data models, 180, 320, 324, 327, 345
- data, *see Data models*
- requirements, 173–200, 173t
 - architecture, defining, 202–214
 - categorization, 204–205, 204f
 - combining structures, 214
 - data models, *see Data models*
 - design options, defining, 214–218, 216f, 217f, 218f
 - detail, need, 205–207
 - DFD, 181–184, 181f, 182f, 183f, 184f, 323, 328
 - matrix documentation, 197–198, 198f
 - potential value and recommend solution, analysis, 218–219
 - process, 173, 174–176, 174f, 176f, 329
 - purpose-based classification, 213–214
 - regulations, industry standards, and corporate policies, 211–213
 - selection of, 198
 - sequence-oriented classification, 208–213
 - stakeholder-based classification, 207–208
 - taxonomy, 202–205, 203f
 - text documentation, 198–200
 - UML family, 184–197; *see also Unified Modeling Language (UML) models*

- validation, 202, 328
- verification, 201–202, 326, 328
- storming, JIT, 48
- Monitoring, risks, 67
- MoSCoW analysis, 138, 310

- N**
- Negotiation, 35, 329
- Network diagram, analyzing risks, 69, 69f
- Neutral facilitators, 305, 312
- Noise, defined, 304
- Nonfinancial costs, 156, 332
- Nonfunctional requirements, 209–211, 304, 318, 326, 329, 339, 340, 345–346
- Normalization, data modeling concept, 180
- Notes, 318
- Novice users, 53–54

- O**
- Objectives
 - business, 153
 - JAD processes, 101
- Object Management Group, 184
- Observations, 87–89, 311, 312, 313, 315, 332
- Observers, 101, 314
- One-size-fits-all approach, 143
- One-way-mirror-type setups, 88
- Onion diagram, 307, 310
- Open-ended questions, 80, 82–83, 84, 93, 311
- Opportunity cost, 222, 329
- Oracle™ Designer, 22–23
- Order processing and shipping, Swede-Mart (case study), 236–237
- Organizational entities, 54
- Organizational model, 330
- Organizational Project Management Maturity Model (OPM3), of PMI, 12, 17
- Organization chart, 160, 354

- P**
- Parking lot, 105, 310, 314
- Participants, in JAD sessions, 96–101, 312
 - business analyst, 98
 - developers, 100
 - facilitators, 96–98, 104–105
 - observers, 101
- scribe, 98–99
- sponsors, 100
- subject-matter experts, 99–100
- user (customer), 99
- Part-time users, 54
- Payback period, 322, 323
- Performance improvements, business analysis, 57–58
- Permissions matrix, roles and, 307
- Perspectives, on business analysis, 26–28
 - agile, 27
 - architecture, business, 27
 - information technology, 27
 - intelligence, business, 27
 - process management, business, 28
- Physical data models, 180
- Plan-driven approach, IIBA, 37
- Planning, business analysis, 58–63
 - approach, 34–49
 - agile approaches, 43–49
 - iterations, 41–43, 41f, 308, 314
 - overview, 36–37
 - selecting and customizing process, 37–38
 - waterfall life cycle, 38–41, 39f
 - governance, 56–57, 307
 - information management, 57
 - manage changes to requirements, 63
 - overview and background, 59
 - performance improvements, 57–58
 - project activities, 60–61, 60f
 - requirements risk plan, 62–63
 - resource plan, 62
 - roles and responsibilities, 61, 61f, 309
 - scope and deliverables, 59–60
- Planning, for test, 224–227, 225f, 226f
- Planning and monitoring, 33–72
 - business analysis, 58–63
 - knowledge area, 306
 - manage changes to requirements, 63
 - overview and background, 59
 - project activities, 60–61, 60f
 - requirements risk plan, 62–63
 - resource plan, 62
 - roles and responsibilities, 61, 61f, 309
 - scope and deliverables, 59–60
 - IIBA view, 34–58
 - approach, business analysis, 34–49
 - governance, business analysis, 56–57

- information management, business analysis, 57
performance improvements, business analysis, 57–58
stakeholder engagement, 49–56
overview, 33–34
requirements, general guidelines for, 64–72
costing, 69–70
estimates, 68
kickoff meeting, 71
laying out tasks, 68–69, 69f
risk management approach, development, 64
risk process, 64–67
risks, assessing, 66–67, 66f
risks, identifying, 64–66, 65f
risks, monitoring and controlling, 67
risks, responding to, 67
tracking and reporting, 70
- Positioning, of BRD, 338–340
removing/adding sections, 339
requirements, types, 339–340
understanding through structured business analysis and standard BRD, 338
- Postcondition state, use case, 190, 326
- Post-It® notes, 58, 115, 119
- Potential value and recommend solution, analysis, 218–219
- Precondition state, use case, 189, 190, 326
- Preparation
- IIBA view, 75
 - interviews, 312
 - JAD sessions, 101, 102–103, 102f, 103f, 107–108
 - requirements package, 126–131
 - customers, 127, 127t
 - development team, 127, 127t
 - executives, 127t, 128
 - external organizations, 127t, 128
 - internal organizations, 127t, 128
 - modeling techniques, 128
 - prototyping, 128
 - text, 128
 - users, 127, 127t
- for test, 301–302
- Prescription Interaction Project, 9, 48, 53, 82, 83, 116, 125–126, 129, 144, 149, 164–165, 195, 212
- Prescription tracking, 174
- Prescription Tracking Project, 106, 182
- Primary actors, in use cases, 192, 323
- Primary users, 53, 343
- Prioritization, requirements, 317, 340
factors, 137–138
techniques, 122–126
 - advantages and disadvantages, 126
 - AHP, 126
 - best practices, 126
 - density dotting, 125–126, 314
 - dollar approach, 123–124, 123f
 - forced-pair ranking, 124–125, 124f, 125f
- Privacy, 210
- Probing questions, 83, 84
- Problem(s)
- documentation, 320, 322
 - solving, business analyst's skills, 28, 303
 - tracking, 317
- Process(es)
- flow, 315
- JAD sessions, 101–107, 101f
conducting, 103–107, 104f
follow up, 107
goals and objectives, 101
preparation, 101, 102–103, 102f, 103f
- models, 173, 174–176, 174f, 176f, 329
- work, cost, 70
- Procurement management, 16
- Product lines, Swede-Mart (case study), 235
- Product-related risks, 63
- Profiling, user, 52–53
- Project activities, business analysis, 60–61, 60f
- Project background, Swede-Mart (case study), 234
- Project charters, 92, 153, 156, 296, 318, 340
- Project Management Body of Knowledge, (PMBOK™) Guide*
- definition of project, 15
 - five process groups, 16–17
 - project budgeting, 16
 - risks, defined, 62
 - for software developers, 36
- Project Management Institute (PMI)
- applicable standards, overview of, 15–17
 - body of knowledge for, 15

ISO certifications, 12
 OPM3, 12
 overview, 5, 8
PMBOK, *see Project Management Body of Knowledge, (PMBOK™) Guide*
 Project manager, 17, 306, 307, 315
 Project rooms, 122
 Prototyping
 advantages and disadvantages, 115
 best practices, 115
 defined, 113
 elicitation and collaboration, 113–115
 evolutionary, 114
 implementing, 114
 for RAD approach, 47
 requirements package, preparation, 128
 throwaway, 114
 types, 114
 Purchasing, Swede-Mart (case study), 235–236
 Purpose-based classification, 213–214

Q

Quality assurance (QA)
 cost, 70
 for stakeholder analysis, 308
 support testing and, 223–230
 acceptance test, 229
 beta/pilot testing, 229
 evaluating customer satisfaction, 229–230
 integration test, 227–228
 planning, 224–227, 225f, 226f
 regression test, 228
 stress test, 229
 system test, 227
 types, 227–229
 unit test, 227
 usability test, 228
 validation *vs.* verification, 224
 Quality management, 16
 Quality-of-service requirements, 209–211
 Questions, types of, 80, 82–84, 93
 closed-ended, 83, 84, 318
 open-ended, 80, 82–83, 84, 93, 311
 probing, 83, 84
 validating, 83, 84

R

Rapid applications development (RAD), 44–45, 44f, 95
 Rational Unified Process (RUP), 43, 189
 Realistic stakeholder expectations, 313
 Receiving, Swede-Mart (case study), 236
 Recordings, 88
 Regression test, 228
 Regulations, requirements architecture, 211–213
 Relationships
 documentation, 179–180, 179f
 identifying, 178
 Reporting
 in analysis phase, 70
 Swede-Mart (case study), 237
 Request for information (RFI), 316
 Requirements
 allocation of, 317, 330, 331
 approval of, 138–139, 317
 architecture, defining, 202–214
 categorization, 204–205, 204f, 306
 combining structures, 214
 design options, defining, 214–218, 216f, 217f, 218f
 detail, need, 205–207
 potential value and recommend solution, analysis, 218–219
 purpose-based classification, 213–214
 regulations, industry standards, and corporate policies, 211–213
 sequence-oriented classification, 208–213
 stakeholder-based classification, 207–208, 209
 taxonomy, 202–205, 203f
 attributes, 309
 communication, 131–132, 132f, 318
 defined, 19, 303
 development, 21
 easy-to-understand, 317
 functional, 209, 305, 318, 339, 340, 345
 gathering
 for customers, 77
 for different categories, 77
 globalization, impact of, 78–79
 risks, 63

- technique, selection, 76–79
techniques, 75t
implementation, 304, 325
life cycle management, 135–139
 approval, 138–139
 changes, assessment, 138
 IIBA tasks, 136–139
 maintenance, 137
 overview, 135–136
 prioritization, 137–138
 traceability, 136–137, 136f
manage changes to, 63
management, 20–21
missed, chance of, 308
nonfunctional, 209–211, 304, 318, 326, 329, 339, 340, 345–346
package, preparation, 126–131, 316
 customers, 127, 127t
 development team, 127, 127t
 executives, 127t, 128
 external organizations, 127t, 128
 internal organizations, 127t, 128
 modeling techniques, 128
 prototyping, 128
 text, 128
 users, 127, 127t
prioritization techniques, 122–126
 advantages and disadvantages, 126
 AHP, 126
 best practices, 126
 density dotting, 125–126
 dollar approach, 123–124, 123f
 forced-pair ranking, 124–125, 124f, 125f
review package, 318
risk plan, 62–63
solution, 209–211, 306
solving problem, 303
stakeholder, 209, 304, 305
transition, 211, 305, 330, 331
waterfall, 38–40, 39f
workshop, 315
Requirements analysis and design definition, 169–219
 IIBA tasks, 172–219
 architecture, defining, 202–214
 categorization, 204–205, 204f, 306
 combining structures, 214
conceptual data models, 180
data models, 177–181, 177f, 178f, 179f
design options, defining, 214–218, 216f, 217f, 218f
detail, need, 205–207
DFD, 181–184, 181f, 182f, 183f, 184f
logical data models, 180
matrix documentation, 197–198, 198f
physical data models, 180
potential value and recommend solution, analysis, 218–219
process models, 173, 174–176, 174f, 176f
purpose-based classification, 213–214
regulations, industry standards, and corporate policies, 211–213
selection of model, 198
sequence-oriented classification, 208–213
specify and model requirements, 173–200, 173t
stakeholder-based classification, 207–208
taxonomy, 202–205, 203f
text documentation, 198–200
UML family, 184–197; *see also Unified Modeling Language (UML) models*
validation, 202, 324, 325, 328
verification, 201–202, 326
overview, 169–172, 171f
traditional models, 171
UML models, 171
Requirements plan, general guidelines, 64–72
activity solutions, for Swede-Mart (case study), 292–295
costing, 69–70
estimates, 68
kickoff meeting, 71
laying out tasks, 68–69, 69f
risk process, 64–67
 assessing, 66–67, 66f
 identifying, 64–66, 65f
 management approach, development, 64
 monitoring and controlling, 67
 responding to, 67
 tracking and reporting, 70
Resource plan, 62

- Responsibilities
 business analysis planning, 61, 61f
 business analysts, 6, 7–8, 7f, 60, 61, 61f,
 142–143, 221, 305
 SDLC, 13, 13f
- Revision log, BRD, 349
- Risk(s)
 analyzing, network diagram, 69, 69f
 assessment, 66–67, 66f
 initial, conducting, 157–158
 BRD templates, 347–348
 categorization, 65, 65f
 elicitation, 307
 initial risk assessment, conducting,
 157–158
 log, component of DSDM, 47
 management, 16
 and mitigation strategies, 292
 plan, requirements, 62–63
 process, requirements plan, 64–67
 assessing, 66–67, 66f
 identifying, 64–66, 65f
 management approach, development, 64
 monitoring and controlling, 67
 responding to, 67
 product-related, 63
 requirements-gathering, 63
- Roles
 business analysis planning, 61, 61f, 309
 business analyst, 6, 7–8, 7f, 60, 61, 61f,
 142–143, 221, 303, 305
 permissions matrix and, 307
 SDLC, 13, 13f
- Root-cause analysis, 28, 307, 319, 324
- Rumbaugh, James, 185
- S
- Safety requirements, 210
- Salesperson, facilitator in JAD sessions, 97
- Sarbanes–Oxley compliant, 201, 310
- Saturn automotive dealers, 51
- Saturn Corporation, 106
- Scope management, 16
- Scope statement, 154
- Scribe, in JAD sessions, 98–99
- Scrum™, 45–46
- ScrumMaster, 46
- Secondary actors, in use case, 192, 328
- Secondary users, 53, 343
- Security
 requirements, 210
 use cases and, 192
- SEI, *see* Software Engineering Institute (SEI)
- Selection
 of model, requirements analysis and design
 definition, 198
 standards, 21–22, 22f
 technique for gathering requirements, 76–79
 for customers, 77
 for different categories, 77
 globalization, impact of, 78–79
- Sequence diagram, 184, 197, 326, 327, 329
- Sequence-oriented classification, 208–213
- Six Sigma standard, 22
- Skilled with advanced tools (SWAT) teams, 45
- Skills, business analysts, 28–31
 analytical thinking and problem solving,
 28, 303
 behavioral characteristics, 29
 business knowledge, 29
 communication skills, 29–30
 interaction skills, 30–31
 tools and technology, 31
- SmartDraw™ Visio™, 22–23
- Software Engineering Institute (SEI)
 CMMI, 12, 37
 risk taxonomy, 202, 203f
 SEI-CMMI and applicability, 19–21
 requirements development, 21
 requirements management, 20–21
 software industry, 8
- Solution evaluation, 221–231
 IIBA tasks, 222
 implementing and supporting, 230–231
 matching solution to needs of customer,
 222–223
 overview, 221
 support testing and QA, 223–230
 acceptance test, 229
 beta/pilot testing, 229
 evaluating customer satisfaction, 229–230
 integration test, 227–228
 planning for test, 224–227, 225f, 226f
 regression test, 228
 stress test, 229

- system test, 227
- types, 227–229
- unit test, 227
- usability test, 228
- validation *vs.* verification, 224
- Solution requirements, 209–211, 306
- Solution scope, 303, 317, 318, 320, 321
 - defining, 152–156, 320, 322
 - assumptions, 153–154
 - business goals and objectives, 153
 - constraints, 154
 - impacted organizations, 154–156, 155f
 - scope statement, 154
- Spiral life cycle, 41–43, 41f
- Sponsors, 50, 51, 71, 80–81, 92, 95, 100, 130, 133, 135, 207, 296, 307, 308, 314, 316, 317
- Sprints, defined, 45
- Stakeholder(s)
 - analysis, 38, 293
 - availability and preference, 315
 - based classification, 207–208
 - for BRD, 127t
 - buy-in, 35
 - categories, documenting, 50, 50f
 - decision making, 317
 - engagement, planning, 49–56
 - communication, 55–56, 56f
 - experienced/novice users, 53–54
 - full-time/part-time users, 54
 - geographical differences, 55
 - internal/external users, 54
 - language and cultural differences, 54–55
 - organizational entities, 54
 - overview, 49–52, 50f, 52f
 - primary/secondary users, 53
 - user profiling, 52–53
 - expectations, realistic, 313
 - getting approval from, 39
 - interaction, complexity of, 310
 - management, 16
 - multiple, 314
 - primary actors in use case, 192, 323
 - requirements, 209, 304, 305
 - Standards
 - BRD, 338
 - IIBA framework and standards, overview, 17–19
 - industry, requirements architecture, 211–213
 - overview, 11–12
 - PMI applicable standards, overview of, 15–17
 - selection, 21–22, 22f
 - tool, comments on, 22–23
 - Stand-up meeting, defined, 46
 - State diagrams, 196, 197f, 326, 327, 329
 - Store buyer, interview with, 238
 - Storyboarding, 115–117, 129, 130f, 314, 326
 - Strategy, Swede-Mart (case study), 233–234
 - Strategy analysis, 141–166
 - business, understanding, 159–160, 303
 - business analysis, 18
 - business models, 160–166
 - entity, 164–165, 164f, 321
 - events, 162, 163–164, 321, 325
 - infrastructure, 161–162, 161f, 321
 - location, 162, 163f
 - organization chart, 160, 354
 - process, 165–166, 165f
 - decision package, preparing, 158–159, 159f
 - defined, 141
 - IIBA tasks, 143–157, 321
 - approach to finding solution, determining, 152
 - assess capability gaps, 150–152
 - assumptions, 153–154
 - business case, defining, 156–157
 - business goals and objectives, 153
 - business need, defining, 145–150, 145f, 146f, 148f, 149f
 - constraints, 154
 - impacted organizations, 154–156, 155f
 - overview, 143–145
 - scope statement, 154
 - solution scope, defining, 152–156
 - initial risk assessment, conducting, 157–158
 - overview, 141–142
 - Strengths–weaknesses–opportunities–threats (SWOT) analysis, 145–146, 146f, 150, 151, 152, 318–319, 322
 - Stress test, 229
 - Structured analysis techniques, 28
 - Structured design, defined, 173

- Structure diagrams, of UML model, 185, 186f
- Structured walkthrough, 309
- Studying
- existing systems, 89–90
 - interfaces, 90–92
- Subject-matter experts (SMEs), in JAD sessions, 99–100
- Success criteria, use case, 190–191
- Summarizer, facilitator in JAD sessions, 97
- Sunk cost, defined, 305
- Supplementary requirements, 209–211
- Supporting, solution, 230–231
- Support testing, QA and, 223–230
- evaluating customer satisfaction, 229–230
 - planning for, 224–227, 225f, 226f
 - types, 227–229
 - acceptance test, 229
 - beta/pilot testing, 229
 - integration test, 227–228
 - regression test, 228
 - stress test, 229
 - system test, 227
 - unit test, 227
 - usability test, 228
 - validation *vs.* verification, 224
- Surveys, 92–94, 310, 312
- Swede-Mart, case study, 9, 233–238
- accounts payable, 236
 - activity solutions for, 291–299
 - distribution center/inventory operations, 234–235
 - industry background, 234
 - order processing and shipping, 236–237
 - overview, 233
 - product lines, 235
 - project background, 234
 - purchasing, 235–236
 - receiving, 236
 - reporting, 237
 - store buyer, interview with, 238
 - strategy, 233–234
- Swim lane diagram, 111, 165, 194, 326, 344
- SWOT (strengths-weaknesses-opportunities-threats) analysis, 145–146, 146f, 150, 151, 152, 318–319, 322
- Systems development life cycle (SDLC)
- defined, 12
 - for IT projects, 13
 - phases, deliverables, and roles and responsibilities, 13, 13f
 - project management processes, 13, 14, 14f
- System test, 227
- T
- Taking tests, 302
- Tasks
- BABOK*, 305, 306
 - input and, relationship, 306
- Tasks, IIBA
- requirements analysis and design
 - definition, 172–219
 - architecture, defining, 202–214
 - categorization, 204–205, 204f
 - combining structures, 214
 - conceptual data models, 180
 - data models, 177–181, 177f, 178f, 179f
 - design options, defining, 214–218, 216f, 217f, 218f
 - detail, need, 205–207
 - DFD, 181–184, 181f, 182f, 183f, 184f
 - logical data models, 180
 - matrix documentation, 197–198, 198f
 - physical data models, 180
 - potential value and recommend solution, analysis, 218–219
 - process models, 173, 174–176, 174f, 176f
 - purpose-based classification, 213–214
 - regulations, industry standards, and corporate policies, 211–213
 - selection of model, 198
 - sequence-oriented classification, 208–213
 - specify and model requirements, 173–200, 173t
 - stakeholder-based classification, 207–208
 - taxonomy, 202–205, 203f
 - text documentation, 198–200
- UML family, 184–197; *see also Unified Modeling Language (UML) models*

- validation, 202, 324, 325, 328
verification, 201–202
requirements life cycle management, 136–139
 approval, 138–139
 changes, assessment, 138
 maintenance, 137
 prioritization, 137–138
 traceability, 136–137, 136f
solution evaluation, 222
strategy analysis, 143–157, 321
 approach to finding solution, determining, 152
 assess capability gaps, 150–152
 assumptions, 153–154
 business case, defining, 156–157
 business goals and objectives, 153
 business need, defining, 145–150, 145f, 146f, 148f, 149f
 constraints, 154
 impacted organizations, 154–156, 155f
 overview, 143–145
 scope statement, 154
 solution scope, defining, 152–156
- Taxonomy
 defined, 202
 requirements, 202–205, 203f
- Team builder, facilitator in JAD sessions, 97–98
- Technical feasibility, 314
- Technical skills, 4
- Techniques
 analogous, 323
 for gathering requirements, selection, 76–79
 for customers, 77
 for different categories, 77
 globalization, impact of, 78–79
 idea-generating, *see* Idea-generating techniques (brainstorming)
 requirements prioritization, 122–126
 advantages and disadvantages, 126
 AHP, 126
 best practices, 126
 density dotting, 125–126
 dollar approach, 123–124, 123f
 forced-pair ranking, 124–125, 124f, 125f
 structured analysis, 28
- Templates
 BRD, 147, 335–352
 appendices, 349
 approach, 342
 approvals, 349–350
 assumptions, dependencies, and constraints, 347
 background, historical, and prior project information, 342
 business-level requirements, goals, value proposition, and benefits, 342–343
 change management process, 348–349
 comprehensive, 336–350
 decision making and approval process, 341–342
 glossary, 340
 information regarding functional requirements, 345
 intended audience, 341
 positioning of, 338–340
 removing/adding sections, 339
 requirements, types, 339–340
 revision log, 349
 risks and risk management process, 347–348
 scope and objectives summary, 340–341
 simple, 351–352
 solution options, 348
 structured business analysis and standard BRD, 338
 technology infrastructure and information architecture compliance, 341
 user and functional level requirements, 344
 user class profiles and key delegations, 343–344
 use cases, 189, 190f
- Temporal event, 328
- Testers, 318
- Test(s)
 acceptance, 229
 beta/pilot testing, 229
 integration, 227–228
 planning for, 224–227, 225f, 226f
 preparation for, 301–302
 regression, 228

- stress, 229
 support testing, QA and, *see* Support testing
 system, 227
 taking, 302
 unit, 227
 usability, 228
 Text documentation, 198–200, 329
 Throwaway prototype, 114
 Time boxing, 45
 Timekeeper, facilitator in JAD sessions, 97, 105
 Time management, 16
 Time zones, concern with global team, 79
 To-be data models, 86, 127, 230, 331
 Tool(s)
 business analysts, skills, 31
 standards, comments on, 22–23
 Top-down estimates, 68
 Total Quality Management, 22
 Traceability
 approval from stakeholders, 39
 ensuring, 310
 forward, 317
 requirements, 136–137, 136f, 211, 211f
 Track drugs, 174, 175, 182, 183, 184
 Tracking
 in analysis phase, 70
 problem, 317
 Training, 210
 Transition requirements, 211, 305, 330, 331
- U**
- Unified Modeling Language (UML) models, 171, 184–197
 activity diagrams, 194–196, 195f
 behavior diagrams, 185, 186–194, 188f, 190f
 interaction diagrams, 197
 main groups, 327
 overview, 171, 184–185
 state diagrams, 196, 197f
 structure diagrams, 185, 186f
 United Nations organizational chart, 354
 Unit test, 227
 Usability test, 228
 Use cases, 185, 186–194, 327, 331
 actors in, 310
 alternative scenarios, 191
- applicable business rules, 191–192
 assumptions and constraints, 192
 author, 192
 back-up and recovery, 192
 candidates, identification, 187
 components, 187–188
 defined, 187
 description, 189
 development, 325
 diagram, example of, 188–189, 188f
 evaluate for potential consolidation of, 187
 exceptions, 191
 extends, 188, 189, 325
 frequency, 192
 gradual expansion of, 193
 identifier, 189
 includes, 188, 189, 191
 key functions and services, 319–320
 main scenario, 191
 postcondition, 190
 preconditions, 189, 190, 326
 scenario in, 191, 329
 security, 192
 strengths, 193
 success criteria, 190–191
 templates, 189, 190f
 version control information, 192
 weaknesses, 194
- User(s)
 business requirements package, 127, 127t
 categorization, 52
 experienced/novice, 53–54
 full-time/part-time, 54
 internal/external, 54
 participant, in JAD sessions, 99
 primary/secondary, 53, 343
 profiling, 52–53
 requirements, 339
- V**
- Validation, 331
 questions, 83, 84
 requirements, 202, 324, 325, 328
 verification *vs.*, 224, 330
 waterfall requirements, 39, 39f

- Vendor assessment, 312
- Verification
 requirements, 201–202, 326, 328
 validation *vs.*, 224, 330
- Version control information, use cases and, 192
- Vertical prototype, 114
- Videoconferencing, 74, 122
- View, IIBA
 elicitation and collaboration, 74, 75–76
 activity, conducting, 75–76
 business analysis information, communicating, 76
 preparation, 75
 results, confirming, 76
 stakeholder collaboration, managing, 76
- planning and monitoring, 34–58
 business analysis, *see* Planning, business analysis
- stakeholder engagement, *see* Stakeholder(s), engagement
- Virtual environments, tools for, 122
- Virtual team, brainstorming for, 121
- Visio®, 115, 194
- Visual thinking, 307
- Volvo Ltd., 4
- W**
- Waterfall life cycle, 36–37, 38–41, 39f
- Wiegers, Karl, 205
- Wiki, 122
- Work breakdown structure (WBS), 58, 59, 60, 60f, 72, 293
- Z**
- Zachman framework, 147–149, 148f, 319, 322