Business Analysis

Best Practices for Success

STEVEN P. BLAIS

The IIL/Wiley Series in Business Analysis

IIL INTERNATIONAL
Institute for Learning, Inc.

Business Analysis

Business Analysis

Best Practices for Success

STEVEN P. BLAIS

WILEY
John Wiley & Sons, Inc.

INTERNATIONAL
Institute for Learning, Inc.

To Sonia: You are on every page and in every word.

Contents

Preface

It is all about change.

There is a problem that needs to be solved. Sales needs support for the new marketing initiative. Human resources (HR) wants the employees to be able to manage their own United Way Fund and other charity deductions online. Marketing needs to change the mailing preferences to allow customers to opt-out of various publications in order to be in conformance with new regulations. The accounts payable system is old and slow and getting more inaccurate by the day. The organization wants these problems solved.

People running the business do not have the time to research, investigate, and determine the best way of solving the problems. Besides, today's solutions require automation, computers, software, and so forth and businesspeople do not do those things. They do not have the expertise. Businesspeople do not want code. They do not want systems. They do not want networks. What they want is a solution to their business problems.

The information technology (IT) department will make it happen. The technology professionals write the software, define and populate the databases, connect the networks, and install hardware. All they need to know is what the business wants done.

Yet, who is defining what will be done to solve the problem? Who defines the solution in such a way that the business can agree with the solution and the technologists can understand what needs to be done to implement the solution? And when the technology is ready for the business, who will make sure the change is made efficiently and the transition from the current to new process is smooth?

The answer to these questions is the business analyst.

Over the past 10 years or so the position of business analyst has found its way into the Human Resources job description catalog of many organizations. It has also earned its own trade group, the International Institute of Business Analysis (IIBA) and its own certification, the certified business analyst professional (CBAP), which is administered by the IIBA.

The role of the business analyst is to solve business problems. Specifying requirements is a critical function of the business analyst, but so are the many other responsibilities a business analyst can and should undertake all of which lead to the successful solution of a business problem.

Business analysis is all about change: changes in business processes, changes in the information technology systems supporting business processes; changes in the way the organization does business. Everything the business analyst does results in some kind of change to the organization. Most of what the business analyst does should be aimed at solving a business problem, and that requires changing the organization from the current situation in which the problem exists to a new process or operation in which the problem has been solved.

First and foremost, the business analyst is a problem solver. Kathleen Barrett, President of the International Institute of Business Analysis, calls the business analyst the ultimate problem solver. The business analyst becomes the go-to person in both the business and development communities when there is a problem. Any kind of problem: political, technical, business, misunderstandings, ambiguities, social, technological, philosophical. Big problems, small problems. Problems that require an IT intervention and those that can be fixed by rearranging the office furniture.

The business analyst accepts the job of proactively understanding what the business problem is and determining the consequences of not solving it and then defines a solution that will remove or ameliorate the problem. The business analyst does this before development starts and then ensures that the solution as built by IT, in fact, solves the problem and does so in such a way that those affected by the problem can use the solution.

By solving business problems, the business analyst is continually adding value to the organization. In fact, all the activities that a business analyst performs add value. The business analyst adds value by:

- Acting as the organizational change agent to improve business processes (Chapter 5).
- Investigating the real problem so that time and energy are not wasted solving the wrong problem (Chapters 8, 9, and 10).
- Providing information to upper-level management so their decision-making can be faster and more effective (Chapters 5, 8, and 10).
- Getting the business managers and process workers to talk directly to the technicians and technologists to reduce time and miscommunication (Chapters 5 and 15).
- Creating an environment where there is an unfettered flow of information between business units and between business and IT that increases quality of overall operations in the organization (Chapters 5, 6, 7, 14, and 17).

- Managing the organization's expectations of the solution so that the stakeholders realistically understand and accept the solution to their problem (Chapters 7, 9, 10, 16, and 17).
- Applying analytical and creative thinking to ensure the organization is making the best decisions and acting on the best solutions to problems (Chapters 5, 8, 12, and 13).
- Assuring the product developed by the solution team solves the intended problem (Chapters 15 and 16).
- Orchestrating the transition from the current business operations to the changed operations so that the organization gains the benefits of the new process as quickly as possible (Chapter 17).

This is a daunting job, filled with challenges and obstacles, both technical and political. And it is also a job filled with satisfaction and personal reward. The business analyst sits in the center of it all, engaging technologists and businesspeople, mediating misunderstandings, defining functions and features, mollifying management, identifying impacts, creating constructive change, and solving business problems.

I have been performing the various roles and activities of the business analyst for 40 years now. I have worked with hundreds of business analysts and have heard their opinions, stories, frustrations, fears, concerns, and questions. This book is in response to them. Their questions, presented as actual quotes from business analysts, appear at the top of each section in which there is an answer. Hopefully, I answered your questions along the way.

My goal with this book is to demonstrate that the business analyst is more than a requirements recorder. The business analyst is a central cog in the successful organization's driving wheel.

The business analyst is the organizational change agent.

The business analyst is the organizational problem solver.

The business analyst is the repository of business process information.

In essence, here are the business analyst's marching orders:

- There is a problem—define it.
- There is a solution to that problem—describe it.
- We need to change the organization to solve the problem—make it happen.

How to Use This Book

While one use of this book might be as a weapon to threaten recalcitrant users into submission, this book can also be used as a guidebook to the wild environs of business analysis. Reading it straight through, from cover to

cover, or at least from page one until the end, you will get a fairly complete description of the overall business analyst's process for solving business problems. You can also use the book to bolster arguments for additional pay and benefits for business analysts or simply to provide supporting information in an effort to establish a centralized formal or informal business analyst group within your organization. However, if you need a quick answer to a question that has been bothering you, the book is also an F&IAQ (frequently and infrequently asked questions) as is described later.

While the main thrust of the book is a description of the business analyst's process for solving business problems, there are also a number of tips, tricks, techniques, and tactics to help to execute the process in the face of sometimes overwhelming political or social obstacles.

The typical business analyst has a finely honed associative memory. It is associative memory that allows the business analyst to relate potential solutions to the business problem and see emerging and existing patterns in the business processes. In deference to that associative memory, the book is littered with sidebars.

Some sidebars emphasize particular points or expand on them.

🔍 Example

Associative memory also allows us to recognize mistakes we have made in the past when we are making them again. This, according to F.P. Jones is the definition of experience.

Throughout the book I highlight tips, techniques, and guerrilla tactics that will serve you in good stead during your business analyst career. Many of the tips are humorous or tongue-in-cheek in nature.

✅ Tip

When you end an information gathering meeting early announce the time you are ending to let people know you are ending early. This way you will be known as someone who ends a meeting on time. If you realize your meeting may be running late, make an announcement about five minutes before the scheduled end of the meeting that "It's about five minutes until the hour and we're about done here. Just a few more questions." If you end ten minutes late most people will still remember the time you stated and have the impression your meeting got out on time.

The Just for Fun sidebars contain fanciful explanations of why things are as they are.

> ## ★ Just for Fun
>
> Whenever we brought changes to the Vice President who was acting as the Change Control Board he would either approve the change or defer it to a later release. He asked what the last scheduled release we had, and schedule it for the next release after that, which at the time was Release 9. When, later on after the first releases of the system were delivered, we began to schedule more releases, he told us to move everything that was in Release 9 out to the next release after the last one scheduled, or Release 12. It was his way of not saying "no" to the business requests for changes to the system. Prior to becoming a Vice President of this telecommunications firm, he has spent years as a consultant in the Washington DC area where he learned how to say "no" without ever saying "no."

Some of the sidebars contain some alternate ways for doing some of the activities you have been performing as a business analyst which might make your job just a little easier, or bring about better results.

> ## 💡 May I Suggest?
>
> Instead of thinking "users" and referring and documenting user activities, needs, wants, etc., think instead "process workers." This enlarges the potential population of people who might be involved in the business process. Users are only involved with the computer and as long as we restrict our views to users we will not see improvements that can be made in processes, especially those improvements that turn process workers into users by automating a part or all of their process activities.

Some sidebars track a case study to show the real-life application of the principles and practices of the business analyst process.

> ### 🌐 Case Study
>
> One of the case studies is an accounts payable system revision. It stars Charlie, the accounts payable voucher entry clerk whose primary goal is to get to Happy Hour on time.

Questions, Comments, and Complaints

Being a business analyst is a complicated job. It is a new profession in many organizations and that newness brings with it confusion, questions, concerns, and the inevitable complaints. Rather than try to guess what the questions are, I asked the business analysts themselves.

The following list represents an abbreviated collection of questions, concerns, and complaints that business analysts have voiced to me over the years. Many of these questions and concerns might have occurred to you as you go about your work as a business analyst. I index the questions to the chapter of this book where the question is answered. This provides a quick reference when the question comes up (again) in your day-to-day activities.

Questions, Comments, Complaints	Answers found in
What is my relationship with the project manager?	Chapter 6
What are the roles and responsibilities of a business analyst?	Chapter 5
What is the connection between requirements and testing?	Chapter 16
How do I know what questions to ask the users?	Chapter 11—The Information-Gathering Session
How can I do it right the first time and avoid rework?	Part Four: The Process
How can I write better requirements?	Chapters 11, 13, and 14
How do we get management and users to cooperate when they refuse to focus on requirements?	Chapter 9
Is it possible to create a common language for IT and business?	Chapters 6 and 7
Is there a methodology or process for business analysts?	This whole book

(continued)

How can I improve the communication between stakeholders and business and developers?	This whole book
Since I'm doing all three roles, what is the difference between the project manager, the systems analyst, and the business analyst?	Chapter 6, Appendix B
Are there any tools for business modeling, and if so which ones should business analysts use?	Chapter 13
How do I negotiate with the business to change their expectations? Or if you can't change them, how do you keep them in line with reality?	Chapter 7
Is there an efficient, effective way to define the requirements?	Chapters 13 and 14
I have to do everything from defining the requirements to coding and testing; how can I effectively be a one-man band?	Chapter 6
How can we make sure there are no surprises at the end when we are delivering the solution?	Chapters 11, 15, and 17
How do we deal with customers who give us the solution and not the problem?	Chapter 11—Interview Issues
What is the best way to objectively define requirements after the boss has given us the solution? What do we do if the real solution isn't his?	Chapter 11—Interview Issues
I deal with both internal and external teams, including offshore developers. How can I make sure all the communications are consistent and effective?	Chapter 5 (Intermediary), 6 (Solution Team), and 15
What's the best way to create the business case? Is it the job of a business analyst?	Chapter 10
Where does the business analyst fit into our software development life cycle?	Chapter 15
We're using agile development (Extreme Programming). What is my role as a business analyst in this situation?	Chapter 15
Is it necessary to provide cost justification, such as an ROI for projects, and if so, how do you do it?	Chapter 10
How do I separate the noise from the true requirements?	Part Four: The Process

(*continued*)

How can I get good requirements when management dictates schedules that don't allow enough time?	Chapter 11
What are some techniques that can be used to work with groups who won't cooperate?	Chapters 7 and 12
What do I do about new requirements that are defined after the project starts?	Chapters 11 and 15
How do I handle the project manager and project team?	Chapter 6
How do I negotiate with the business to change their expectations?	Chapter 7
How do we handle changes after getting sign-off on a hundred-page document?	Chapter 15
The business analysts are tasked with testing the results of the development efforts. We are not given much advance warning. Then when we use the requirements as a guideline to what we expect the system to do, it's all different. The technical team has made changes and we don't know what the system is supposed to do. How can we test it on behalf of the users if it isn't what the users asked for anymore?	Chapters 15 and 16
I have been a systems analyst for over five years; how do I transition to my new job as business analyst?	Chapters 3 and 6
Communication with the developers is not very satisfactory. They have no respect for what we do.	Chapter 6
Over-commitment—management is trying to do too many things without evaluation or prioritization.	Chapter 7
How do I explain to my kids what a business analyst does?	Part One: The Problem Solver
I transitioned from system analyst to business analyst. Will be technical background help me or hurt me?	Chapter 3
How does the time spent in business process modeling help me? Do I need to know how to do all the different types of models, like entity relationship diagrams?	Chapter 13

How do I get the business to give us information?	Chapter 11
Is there a holistic view of requirements and testing?	Part Four: The Process
There are last minute changes made to the releases which are done directly with the project team. When this causes the delivery to be delayed or there are impact problems, the business analysts are blamed.	Chapter 15—Checkpoint Charley
There is no single point of responsibility for documenting and maintaining all the communications between business and technical teams about the project and requirements.	Chapter 5—Intermediary
How can we convince the users that we do more than prepare and maintain documents?	Chapter 1 and Postscript
There are user meetings every month, but the business analysts are not allowed to attend since we represent IT and the meetings are for the business.	Chapter 7
Are there any overall guidelines that will assist business analysts in doing their job successfully?	This whole book
What can I do to increase collaboration among all the parties in the solution development effort?	Chapter 5—Diplomat
Why is there always such a gap between the user requirements and the delivered product?	Chapters 8, 9, and 15
How can we make successful changes to the processes without encountering so much resistance from the users?	Chapters 12 and 17
I feel like we are an afterthought. Is there really a business analyst profession?	Chapters 2, 4, and Postscript
What is the difference between the "what" requirements and the "how" requirements?	Chapter 14—Anatomy of Requirements
Who defines acceptance test cases? Who executes acceptance test cases?	Chapter 16
How do we convince the customer to do something different, such as another approach?	Chapters 7, 11, and 12

Acknowledgments

Thanks most of all to my wife, Sonia, who put up with vacations spent in writing and rewriting to hit deadlines and missed appointments and engagements, late nights and constant travel that come with the jobs that generated the ideas and examples included in the book.

Finally, thanks to all the business analysts everywhere who through their persistence and hard work are creating the profession that will be at the center of every successful business in the twenty-first century.

International Institute for Learning, Inc. (IIL)

With operating companies all over the world and clients in more than 150 countries, IIL is a global leader in training, consulting, coaching and customized course development. Our core competencies include: Project, Program and Portfolio Management; Microsoft® Project and Project Server; Business Analysis; Lean Six Sigma; PRINCE2®; ITIL®; Leadership and Interpersonal Skills.

Using our proprietary Many Methods of Learning™, IIL delivers innovative, effective and consistent training solutions through a variety of learning approaches, including Virtual Classroom, Traditional Classroom, Simulation Training, Interactive On-Demand Training and a blended approach. Our roster of international trainers is comprised of elite professionals whose industry experience is enhanced by their practical classroom expertise. To learn more please visit www.iil.com or email Learning@iil.com.

The Problem Solver

The business analyst solves business problems. The business analyst adds value to the organization. The business analyst does this not by defining a set of requirements so that a solution development team, at the behest of a technical project manager, can use them; nor does he run interference between the business wonks on one side and the technology geeks on the other. Being a business analyst means one is in the center of change in the organization, and that is a dangerous place to be without a map, or at least a good plan of action, or perhaps a better escape route.

The problems that face today's organizations and the fast pace of business change can seem overwhelming to one who is charged with solving those problems and keeping up with the pace. Being in the center can give the business analyst the uneasy feeling that BA stands for Blame Attractor.

The whole process of solving problems and implementing solutions, especially technological solutions, can be made easier by adopting a systematic approach, one that can be used each and every time and one that has gained credibility through successful use in the past. Thus we have the systems approach to solving business problems. And at the center of this approach is the business analyst.

What Is a Business Analyst?

The job market is undergoing a shift in requirements from general computing knowledge and programming skills to those of inter-disciplinary domain knowledge and integrated application development and problem-solving skills.

—Jiming Liu

What is a business analyst? Why is such a position necessary to organizations? Is the business analyst simply a middleman between the technologists and the businesspeople, acting as a go-between, translator, and conduit? Or is there some larger, more important role being played in the center of the organization? This chapter explores what makes a business analyst and what a business analyst does for the organization. It also takes a look at the potential of the position and the direction in which the business analyst role is evolving.

The Business Analyst in Context

There is a new position in the corporate hierarchy. A purebred technologist or an entirely business-oriented worker cannot fill this position. It is not management level and does not possess authority; however, it is a key contributor to most of the successful IT-related changes in an organization. Those occupying this position are fully versed in how to increase productivity, lower costs, and comply with regulations from both the business and technology perspectives. They can look at any problem from the perspective of the entire organization to determine the impacts, positive and negative, of any proposed change. They are adept at fashioning solutions to business problems, generally using computer technology. This position is the business analyst.

Since the first time a computer was used to support a business process, there has been a need for someone to talk to businesspeople. Until there is a time when businesspeople can solve their problems directly with the computer without needing a technologist to design the programs and change the code, there will be a need for someone to help businesspeople define the problem and describe the solution to the technical people who solve it.

"I've been a business analyst for twelve years. My new boss doesn't have a clue about business analysts. He thinks 'business analyst' is just a new term for requirements collector. Can you tell him the value of business analysts? He won't believe me."

The business analyst position is relatively new in the organization. Many organizations do not have a defined business analyst position as yet. The reality is, though, that the business analyst is not a new role to the organization, but rather a role that has been played since the first business owner challenged his staff to come up with a more efficient way to produce wheels. While there may not have been an official position in most companies called business analyst, for years the role has been performed by other positions in the organization, such as project manager, systems analyst, and business manager.

What Is It All About?

"Can you tell me in a nutshell, like an elevator pitch, what it is that a business analyst does so I can tell my mother-in-law?"

In Version 1.6 of its *Business Analysis Body of Knowledge* (BABOK), the International Institute of Business Analysis (IIBA) has the following definition of the role:

> A business analyst works as a liaison among stakeholders in order to elicit, analyze, communicate and validate requirements for changes to business processes, policies, and information systems. The business analyst understands business problems and opportunities in the context of the requirements and recommends solutions that enable the organization to achieve its goals.[1]

In 2009, the IIBA updated its definition to "A business analyst is any person who performs business analysis activities, no matter what their job title or organizational role may be." Business analysis activities involve "understanding how organizations function to accomplish their purposes, and

defining the capabilities an organization requires to provide products and services to external stakeholders. It includes the definition of organizational goals, how those goals connect to specific objectives, determining the courses of action that an organization has to undertake to achieve those goals and objectives, and defining how the various organizational units and stakeholders within and outside of that organization interact."[2]

The British Computer Society proposes the following definition of a business analyst:

> An internal consultancy role that has the responsibility for investigating business systems, identifying options for improving business systems, and bridging the needs of the business with the use of IT.[3]

These authorities have different slants on the business analyst job: analyst, liaison, communicator, internal consultant, improver of business systems, and business problem solver. Putting it all together, the business analyst is an agent for change in the business, summoning the forces of technology to make changes in the organization, solving problems, and improving processes, thereby increasing the value of the organization.

The Role of the Business Analyst

"I'm a project manager and it sounds like I have been doing the business analyst's job for quite a while. Is that possible? Should I get two salaries?"

Over the years the work of business analysts evolved first into a role and more recently into a position in the organization. Where there is not a business analyst position, the role has been played by other positions, such as the IT project manager or a business line manager, on a part-time or temporary basis. In some organizations, it is divided among several positions, such as requirements engineer, quality assurance analyst, quality control specialist, product owner, project manager, business champion, software configuration manager, and so forth. Organizations are now realizing that the majority of IT project failures occur because no one person took on the role of business analyst, but still there is no true agreement on what that role should be. This section explores many of the options.

The following is a quote from an East Coast utility company's internal document entitled *Business Analyst Handbook*. Note the emphasis on the business analyst's roles:

> At [company name], the business analyst serves many functions, from operational business support of a business area to deep involvement

in software development projects. The business analyst's role changes based on the customer area he or she is supporting. This situation exists because the expectations for a business analyst are customer driven. A business analyst can be focused on a business area supporting many applications and processes or a single large application (such as an enterprise application) or they may possess extensive knowledge in a particular business area process and support technology associated with that process. Whatever the role, the business analyst must possess a wide variety of skills and knowledge ranging from strong relationships, excellent communication skills, problem solving, facilitation, quality assurance techniques, presentation skills, and analytical/critical thinking. Sprinkled in with all these skills, it is important for the business analyst to have a surface understanding of the technological infrastructure (network, applications, software and hardware) that supports his or her business area.[4]

The Business Analyst in the Center

No matter how you look at it, the business analyst's role is in the center. As shown in Figure 1.1, there are three communities that the business analyst must deal with throughout any project and thereafter.
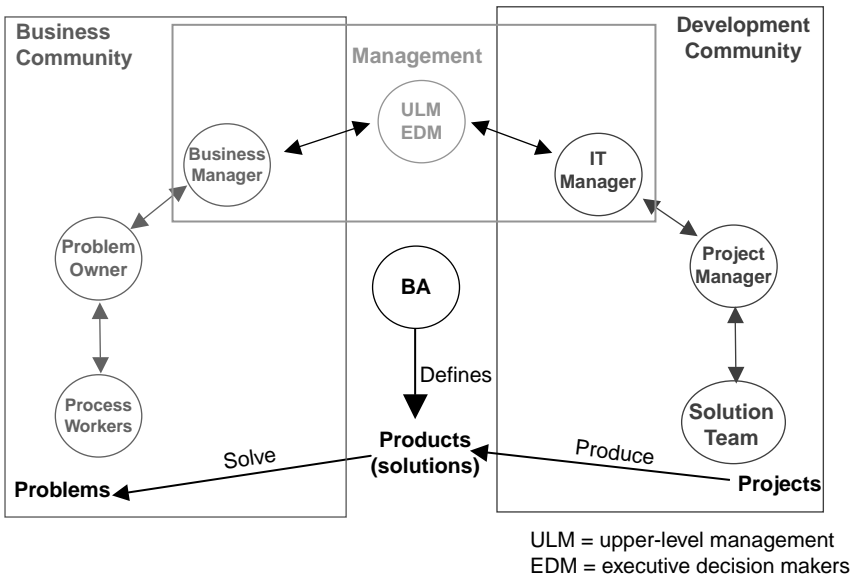
FIGURE 1.1 The Business Analyst in the Center

The business community represents the slice of the business that is involved with the problem to be solved. It might be a large slice, such as accounting, or it might be a small slice, such as the collections department. Generally this business slice represents the problem domain.

The business manager is the highest-ranking person in the organizational hierarchy directly associated with the business area. For example, when a problem exists in the collections department, the business manager might be the manager of the collections department. When a problem exists in accounting, replacing the accounting system for example, the business manager might be the CFO.

The problem owner is the primary point of contact for the problem. The problem owner is the person who has authority to seek a solution to a perceived problem in the business area. The process worker is anyone who actually works with the system or business process in question as a part of his or her daily job. The term user refers to a subset of process workers, namely those who actually use a computer system and put data into a system, extract information from the system, and manipulate the information within the system. I suggest instead the term *process worker* to expand the business analyst's view to include those in the business community who are involved with the overall business process being improved, but who are not necessarily users of a computer system. This helps to keep our focus on the business rather than the technology.

The business community has problems. There are changes in government regulations to deal with, new products introduced by the competition to keep up with, new markets to break into; there is expansion of sales and support, mergers, acquisitions, divestitures, and personnel turnover. There are old legacy systems that cannot cope with the new marketplace and product lines; and there are the inevitable defects that crop up and small changes to be made to the computer system. When the business community can solve these problems, it does. Because of the impact of computer technology on every aspect of the business for most organizations, the business community generally needs the help of development community personnel to solve the business problem. In fact there are many times that the development community looks on the business community as nothing but one big problem. This is good. If the business did not have problems, the development community would not have work.

The development community in Figure 1.1 represents all of IT. So the IT management circle is the highest-ranking person on the IT side, such as the CIO or vice president of management information systems (MIS).

The job of the development community is to execute a successful project. A successful project is defined as being within budget, meeting the scheduled deadline, and delivering everything that was promised for that budget and schedule. Except for ongoing operations, everything on the

development side is a project. From a project perspective, the team is not concerned with whether the result of the project actually solves the problem, only that the project is a success. The project manager and solution team rightfully assume that the business has done due diligence and determined that the product to be developed is necessary and will provide a benefit to the organization. The solution team's job is to make it happen within the budget and timeframe.

So here is the situation: The business community has a problem and the technical community creates a product purportedly solving that problem, and there is no correlation that the problem is solved until the project is done, if then.

Perhaps the coordinating function is upper-level management. The management box across the top of Figure 1.1 represents upper-level management and executive decision makers up to and including the CEO and board of directors.

Upper-level management charts and monitors the strategic direction of the organization. Since projects are tactical, upper-level management is not typically concerned with the details of projects. When upper-level management does get too involved in the project details, we have a word for it: micromanagement. Process workers also have a word for the upper-level managers who do this sort of thing, but that word is better left unsaid.

So we still have a situation. The business community has a problem, one of a tactical nature, and the development community has a project, also of a tactical nature. This project is designed to produce a product. That product should be the solution to the business problem. However, there is no formal correlation between project and problem. The solution team assumes that the business has determined why the project is needed and what value the results of the project will provide to the business. The business assumes the solution team is going to come up with a solution to their problem and that it should be obvious why the project needs to be done and what the results have to be.

So who will verify that the result of the project—the product—completely solves the business problem? The role that ensures the results of the project solve the business problem is the business analyst. That is why the ideal position in the organization for the business analyst is in the center, unaligned with either community. The business analyst independently evaluates the business problem and specifies the solution for the solution team and then makes sure that the solution solves the problem it was intended to solve.

Business Analyst Focus

The business analyst focuses primarily on the business. In some cases, this means that the business analyst is not involved with IT at all. For example, the business analyst may be involved in rearranging job descriptions and

reorganizing manual tasks as part of a process improvement effort, assisting upper-level management in determining business strategy, or gathering the information and performing benchmarks for requests for proposals (RFP). Regardless, the focus is always on the product, the solution to the business problem. The ultimate goal of the business analyst is to solve that business problem, nothing less. When technology is involved the business analyst is a member of the solution team, but is still focused on the solution. In many situations, the business analyst is the only one so focused.

"I'm not really sure of my job duties as a new business analyst. What is a business analyst supposed to be doing? What do other business analysts in the industry do?"

The truth is that the industry has not really come up with a standard definition of what a business analyst does, even with the definitions in the IIBA's *Business Analyst Body of Knowledge* and other sources. This is because business analysts have come from both the technical and business sides of organizations and the role is still evolving (see Chapter 5 for a view of the various roles of the business analyst), so there has not been coalescence on a single definition. Here is an analogy that I think captures the essence of the business analyst: the business internist.

The Ideal Business Analyst

"Can you tell me what to expect when I start my job as business analyst next week? What do management and everyone else expect from me?"

Table 1.1 provides a generic job description for the ideal business analyst broken down into task-related categories.

TABLE 1.1 The Ideal Business Analyst

Problem Analysis and Solution Definition	General Communication
Determines the actual problem to be solved in the organization.	Both facilitates and moderates meetings.
Understands the business issues and challenges of the organization and industry.	Delivers informative, well-organized presentations.
Identifies the organization's strengths and weaknesses and suggests areas of improvement.	Understands how to communicate difficult/sensitive information tactfully.
	(continued)

TABLE 1.1 (*Continued*)

Problem Analysis and Solution Definition	General Communication
Reviews and edits requirements, specifications, business processes, and recommendations related to proposed solution.	Possesses enough understanding in technical disciplines to be able to converse intelligently with solution team.
Documents the solution to the business problem in a form approvable by the business, acceptable to the solution team, and understandable to management.	Mediates conflicts between business and the solution team and different business units being impacted by the solution
Pushes creative problem solving beyond the boundaries of existing organizational practices and mind-sets.	Generates enthusiasm for the product among product stakeholders and solution team members
Identifies areas for improvement in internal processes and suggests potential solutions.	Facilitates decision making among organization executives
Product Delivery	**Product Quality Assurance**
Receives input from managers and appropriately and accurately provides comments/feedback.	Evaluates requested changes from the business and communicates needed changes to development team.
Communicates non-technical product and business standards and constraints.	Ensures product issues are identified, tracked, reported on, and resolved in a timely manner with both the solution team and the business.
Facilitates the business-community transition from current problem state to solution state.	Leads and/or participates in acceptance testing efforts.

Product Stakeholder Relationship

Corresponds effectively with the business to identify needs and evaluate alternative business solutions.

Identifies and manages product stakeholder expectations effectively.

Ensures that the organization will be ready to accept and affect the change.

Conducts effective product evaluations to ensure the problem is being solved in the business environment.

This is quite a responsibility for a business analyst to undertake. It is all part of a holistic view of the organization, the business problems, and the IT solutions.

Creating positive change for the organization is the essence of the business analyst. Problem solver, communicator, facilitator, analyst—the business analyst works in the center of the organization improving

processes, clarifying communications, investigating problems, producing solutions, and adding value to the organization.

Last-Liners

Reviewing the list of jobs a business analyst does from Table 1.1 there seems to be very little in the organization that the business analyst does not do. I did not include in that list random tasks mentioned, such as prepare for executive meetings and make coffee. The business analyst position seems to be the epitome of what we call last-liners, referring to the last line on most job descriptions, which says something like "and any other activity or task required by management." Last-liners are those whose entire workday is filled with tasks and activities not listed on their job description but are covered by that last line.

So is the business analyst really the new kid on the block? Has there been a sea change in business and IT that has resulted in the creation of this position? No. Actually, the role of business analyst has been around for centuries, perhaps as long as there has been business or at least accounting for business. Business analysts are not quite the oldest profession, but the position actually predates the modern computer, giving further support to the contention that business analysts solve business problems rather than write software requirements. Don't believe it? The next chapter describes a bit of the evolution of the business analyst and identifies some of the famous and infamous business analysts throughout history.

Notes

1. International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge,* version 1.6 (2006), page 9.
2. International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge,* version 2.0 (March 31, 2009), page 4.
3. Debra Paul and Donald Yeates, *Business Analysis* (British Informatics Society Ltd, April 2006), 4.
4. Excerpted from internal corporate business analyst handbook for an East Coast utility company, published 2006.

The Evolution of the Business Analyst

It is our responsibilities, not ourselves, that we should take seriously.
—Peter Ustinov

Every profession has its history and heroes. The medical profession's progression from Hippocrates to Gregory House is well documented. The legal profession can point to Clarence Darrow and Daniel Webster, among many others. These luminaries stand as models and beacons of their profession. Defining the origin of the business analyst and tracing the profession's history is much more difficult. The need for a business analyst would exist without computers or information systems, although the present-day business analyst role has roots in the evolution of business computer technology. The role is a melting pot of professions and disciplines, technology and intuition, solitary analysis and personal interaction, engineering and business practice. Let's look at the history of the business analyst and some of the influences on the profession.

The Business Analyst Hall of Fame

There have been people playing the role of business analyst for centuries now. Perhaps our first recorded business analyst might be Adam Smith, who documented the business process of manufacturing a straight pin to make a point about separation of skills. Smith's assertion that the economy and all business is ruled by the invisible hand of self-interest has great influence on

the business analyst's work, as stated in this oft-quoted passage from *Wealth of Nations*:

> It is not from the benevolence of the butcher, the brewer, or the baker that we expect our dinner, but from their regard to their own interest. We address ourselves, not to their humanity but to their self-love, and never talk to them of our own necessities but of their advantages.[1]

This may be the original WIIFM (What's in It for Me?) statement. As we will see, the business analyst who understands this philosophy in business will have a much easier time obtaining his or her own goals.

Another who might be placed in the Business Analyst Hall of Fame would be Herman Hollerith, who solved a compelling business problem in the late 1800s. The U.S. Census Bureau is legally mandated to count the people in the country every 10 years. The 1880 census took seven years to complete and by 1890, the country had grown so much more it would have taken more than 10 years to conduct the census. Hollerith borrowed an idea from the textile industry and created a mechanical counting device based on cards with holes punched in them. While the machine was a computing device (thus qualifying Herman for the Computer Hall of Fame as well), Hollerith was truly a business analyst solving a business problem using computer technology. Hollerith, incidentally, founded a company called the Computing Tabulating Recording (CTR) Corporation. You have probably not heard of the company, except that later on, when the president of CTR was Thomas Watson, the company was renamed International Business Machines (IBM) Corporation.

Also in our Hall of Fame is Frederick Winslow Taylor, who solved business productivity problems with innovative workforce management approaches. Taylor was the first to apply systematic observation and study to the workplace. He also believed that by analyzing work activities and flow, the one best way to perform the work could be determined. Business analysts today use systematic observation and analysis to determine the one best way to solve the business problem they are assigned to solve. They routinely find the best practice wherever it exists, decompose tasks into essential parts, and remove things (e.g., operations, activities, etc.) that do not add value.

A special niche is reserved for Sherlock Holmes, Arthur Conan Doyle's fictional detective. He established, fictionally of course, the basics of scientific investigation and examining all the available data before coming to a conclusion about a solution. We discuss the advice that Mr. Holmes makes to the business analyst throughout the book. We might also find nearby a plaque with the face of Mark Twain, who provided significant guidance in gathering data and assembling facts from a reporting perspective. For

example, he is reputed to have said, "Gather the facts first. You can distort them later." This is sage advice for the practicing business analyst. We see more of his counsel later when we discuss elicitation and investigation.

Quality gurus such as Armand Feigenbaum (Total Quality Management or TQM), Phil Crosby (*Quality Is Free*), and William Edwards Deming (Theory of 14 points, among others) might also be added to the list of people who have significantly influenced the business analyst profession. Feigenbaum's book *Total Quality Control* established a holistic approach to instilling quality in the workplace. The business analyst uses a similar holistic approach to solving business problems.

We would have to include Alex Osborn, an advertising manager from Buffalo, New York, who was one of the founders of BBDO (Batton, Barton, Durstine, and Osborn), one of the largest and best-known advertising companies in the world. An advertising manager in the Business Analyst Hall of Fame? Yes. Alex Osborn developed the brainstorming method and other techniques used often by business analysts, as well as other creative thinking and problem solving methods.

Influencers of more recent vintage are Michael Hammer and James Champy, who coined the term business process re-engineering (BPR) in the seminal volume, *Re-Engineering the Organization*; Bill Smith of Motorola, who brought Six Sigma to the attention of business and IT; and others whose words of wisdom and guidance to the business analyst are liberally sprinkled throughout this book.

You might notice that the honorees in the Business Analyst Hall of Fame are primarily non-IT people. Where are John Van Neumann, father of the computer, and Vint Cerf, father of the Internet, and Tim Berners-Lee, father of the World Wide Web? How about Bill Gates, Larry Ellison, and Steve Jobs? Certainly, these illustrious gentlemen would grace any technological or IT Hall of Fame—but not our Hall of Fame. These titans of the IT industry gave us better means to solve problems, not the solutions themselves. Some might even argue that their contributions have created more obstacles to solving business problems because the technological advances tend to move our focus from the business to technology. Our hypothetical shrine is for luminaries who have developed and practiced methods for solving *business* problems with technology and improving the organization's business processes.

Where It Began

Back in the 1960s when I started in the computer business as a programmer, there were no business analyst positions. In fact, there were few systems analysts or software architects, or any other intermediary role. We were all programmers and we met with the businesspeople directly. Since user

nowadays refers to someone who interacts directly with a computer system, we cannot even say there were users at the time. There were employees performing work that might be done better and faster on a computer. The user interface was limited to what was punched onto Hollerith cards (called 5081 cards or simply punched cards) and the reports that were generated by the computer. The people we talked to were cooperative, although somewhat skeptical, and certainly a bit fearful. Despite that, we communicated fairly well. We paid no attention to the business processes these computer systems supported; that was not our job.

Most of the programmers and computer technicians at that time came from engineering and mathematics curricula in college or from the fledgling Computer Science departments, and were not skilled in human interaction and tact, much less business. Programmers and other computer technicians lived behind the glass walls of computer rooms whispering incantations over their machinery and posting large "Keep Out" signs on the doors. A sizable contingent of the data processing populace believed that computer technology was a science or engineering discipline and that producing reports for business was a sideline, something to be tolerated. This, of course, led to severe misunderstandings between the business community and the programmers.

As a result, data processing departments created the programmer analyst role. Technicians given this role talked to the business and translated the business requests into program specifications for the programmers. This excused programmers from having to converse directly with the businesspeople who were, in turn, much relieved. Interestingly, those assigned the role to talk directly to the business tended to be graduates of vocational schools teaching computer programming rather than the computer scientists matriculating from colleges. Colleges and universities at that time did not teach business courses to computer scientists and engineers, and the graduates from the programming schools came from professions and businesses already possessing insights into how business worked.

As technology became pervasive in business and government organizations, it became increasingly difficult to merge the demands of the organization and the technological advances of computers. Computer scientists added more complexity to the technology with newer, faster computers and peripheral devices. Businesspeople then discovered the many things computers could do to make their work easier, and started demanding computer departments to harness computer power for business processes. This, in turn, caused the computer scientists to build bigger, better, and faster computers to keep up with the business demand, which caused the business to create new demands for the better and faster computer technology, and around it went.

Unfortunately, the advances in both arenas did not proceed in the same direction. As a result, the programmer analysts became too technical for the business and the business became too complicated for the programmer analyst.

Then the data processing department, as it was called then, invented the position of systems analyst to reconcile the divergent areas. The systems analyst was supposed to talk to the business and arrive at a technical solution for an entire system consisting of hardware, data, and software (networks were not an issue then), which was then turned over to the programmer analyst to write program specifications, which were then turned over to the programmer to code. However, systems analysts still did not concern themselves with business process. They focused solely on computer support of those business processes.

On the business side, senior managers drew straws to see who would be the one to have to explain their needs to the computer guys. The user interface at that time was a simple terminal on which questions were displayed. The responses were entered in a scrolling, sequential fashion in gray characters on a green background. Secure in the notion that we knew best what the users wanted, we designed the systems without consultation with those users. We occasionally consulted with the managers of those users, but they rarely understood exactly what their people did. Because we systems analysts ignored the impact of the systems we were designing on the business processes in place, the newly minted users of the systems were forced to change everything they were doing and use keyboards instead of paper and pen and then wait for hours, if not days, to see the results printed on 11×14-inch green bar (striped) paper.

⭐ **Just for Fun**

This was also when the first occurrence of the now legendary Stupid User Error happened. The exchange went something like this:

> Business manager: The system crashed when the user entered the date wrong.
>
> Programmer: Well, if the user entered the date right, it wouldn't have crashed. My code works. Tell your users to enter the date right. (To himself) Stupid users!

Information Systems

Even though it was still a back-room function in the 1970s, data processing was coming into its own as a viable department in the organization. With the newfound power it was experiencing, data processing renamed itself

information systems (IS) or the management information systems (MIS) department, which elevated it to a larger influence in the business.

Those in IS/MIS making direct contact with the businesspeople were still the same technically-oriented engineers who were not familiar with the business and, for the most part, did not care to be. These technicians were too busy dealing with rapidly evolving computer technology that went from second to third generation computers and trimmed down from large, room-sized mainframe computers to mini-computers, then micro-computers, and then personal computers. The software industry expanded and changed, creating new programming languages, moving from second-generation languages to third generation to fourth generation, and then to visual development environments. Software development progressed from the controlled, linear approaches typical of engineering with structured design, to object-oriented analysis and design, to iterative and incremental approaches, to agile software development methods. Coders became programmers and then became developers. The computer technician became a technologist and had to become more technology focused, just to stay viable in the industry. And the business started selling internationally, dealing with regulations of multiple nations and varying cultures. As we entered the last decade of the twentieth century, businesses started merging and acquiring, growing bigger and more complex, becoming global in their scope.

The Rise of the Business Analyst

Businesspeople found themselves increasingly unable to communicate with the staff working in information technology (a term more reflective of the prevailing attitude). Business management was forced to delegate members of its staff to meet with the IT people and explain what the business needed, and saw its employees relegated to the role of user. The thinking went like this: "He uses the computer, therefore he is a user, therefore he knows what he wants to use the computer for, therefore he can explain it clearly to those technical people. Better him than me." Usually the one designated was a super-user (an IT term) who was not cowed by the computer or technology, and was the user other users called when they had computer problems and were too afraid or embarrassed to call in the nerds. (Nerd is a term invented in 1950 by Dr. Seuss in the book *If I Ran the Zoo*. The term evolved over the years to connote a person more interested in esoteric and technological activities than in social interactions. The word described the stereotypical computer programmer back in the early days of the industry.)

This super-user was, most likely, the first official business analyst but probably did not have that title. We do not have a date or name or location to commemorate the first time a businessperson was assigned the job of

crossing the no-man's land between business and IT to solve a problem. We do not even know whether that representative was issued a white flag to wave.

★ Just for Fun
_____

How it might have happened . . .

Anne, an accounts receivable supervisor, went to the VP of Finance saying that she was spending all her time with the computer guys discussing new systems and was not getting her accounts receivable work done. The VP of Finance decides to create a new job for Anne, and made her a full-time business analyst.

Or, on the IT side, David, the developer, complained to the Director of Software Development that he did not have time to code or design because he was spending too much time in meetings trying to find out what he was supposed to code and design. The Director gave him the choice of continuing to meet with the business or go back to coding and designing. When David chose the former, the Director assigned him the title of business analyst and banished him from the technologists' lunchroom.

IT also recognized the communication problem. Perhaps IT departments saw the business sending representatives over the wall to talk with them and decided that IT needed its own representatives. Perhaps the IT department was tired of having all its systems analysts and programmers talking directly to the users, which caused all sorts of havoc, and decided to funnel the communications to the few technologists who could string together a meaningful English sentence and possessed enough tact to keep from insulting the other parties for at least the first half hour. In any case, IT began assigning and appointing the role of point person to meet with the business and discuss what really had to be done to solve the problems of the business.

The actual term *business analyst* had been used in business in a different context for a long time. The title referred to someone who analyzes business processes or activities to discern better ways of operating and making a profit, such as analyzing the competition's product line to locate holes that can be filled by the analyst company's products. Another type of business analyst also analyzes other businesses and reports his or her findings to stockholders, investors, financial institutions, market researchers, and so on. These business analysts are found on Wall Street and in companies like

Gartner and Forrester. Typically, neither of these types of business analysts have direct interaction with a software development team.

Throughout the history of business there has been a need to search for the solution to business problems. Businesspeople, inventors, technicians, managers, company founders, innovators, and workers have all sought to solve new and recurring business problems, some achieving Hall of Fame status with their efforts. Over the years a key role has emerged whose specific purpose is to solve those business problems: the business analyst.

The Business Analyst Position

Think as you work, for in the final analysis, your worth to your company comes not only in solving problems, but also in anticipating them.

—Tom Lehrer

"My position is business systems analyst. Is that the same thing as a business analyst?"

There are a multitude of titles describing the business analyst. Each organization seems to have its own view of what a business analyst is and does. I worked in some organizations where the definition of system analyst is indistinguishable from that of business analyst. I also worked with business analysts who shared the same title but do completely different jobs. A large U.S. government agency had about 30 business analysts all working in the same large room in cubicles. The agency assigned about half of the business analysts to define the requirements and assigned the other half to do testing of completed systems against those requirements. The testers never defined the requirements and the requirements group never tested. They were all called business analysts.

In a health insurance company in New York, the business analysts are referred to as customer champions. A New England insurance company assigns the business analyst role to an IT function called business relationship manager, whose job it is to keep the business informed and satisfied with the work that IT is doing. In a large U.S. federal government agency, the business analyst is drafted from the line managers and wears the business analyst hat for the duration of the project before returning to their regularly assigned duties. In one of the larger U.S. banks, the business analyst position is split into technical business analyst (TBA), who focuses on specifying the software requirements, and business business analyst (BBA), who creates the business requirements.

In fact, without too much of a stretch, sales support people might well fall within the overall category of business analyst. They bridge the gap between external customers who have problems and may be in need of the products the organization is offering, and the solution represented by those products.

The bottom line is that **the role of the business analyst is the customer-facing member of the solution team**, or, looking at it from the other direction, the representative of the business on the solution team. In all cases, the business analyst, under whatever guise or title, determines the business problem, analyzes the situation, identifies the best solution, and ensures that the solution solves the problem in the business environment.

The Modern Analyst Forum Web site for business analysts and systems analysts lists the following variations for business analyst roles, each variation defined in detail by the Modern Analyst Forum:

- Business analyst (general)
- Business process engineer/analyst
- Data analyst/modeler
- Product manager/functional architect
- Requirements engineer
- Systems analyst
- Usability/UX professional

The Web site goes on to list related roles:

- Designer/architect
- Developer
- Quality Assurance analyst/tester
- Project manager
- Technical writer[2]

The Business Analyst Profession

The most likely way for the world to be destroyed, most experts agree, is by accident. That's where we come in; we're computer professionals. We cause accidents.

—Nathaniel S. Borenstein

"Are there any business analyst organizations where I can meet other business analysts?"

One indication of the maturity of a profession is the existence of professional organizations devoted to the advancement of those members of that profession. Business analysts have a range of professional organizations that relate to some of the roles business analysts play, predominant of which is the International Institute of Business Analysis (IIBA) and the Project Management Institute (PMI). Business analysts become members of other professional societies and related organizations, such as the American Society for Quality (ASQ), the Institute of Electrical and Electronics Engineers Computer Society (IEEE), the International Standards Organization (ISO), the British Computer Society (BCS), the Institute of Analysts and Programmers (IAP) in the United Kingdom, the Australian Business Analyst Association (ABAA), and others.

The International Institute of Business Analysis (IIBA) describes itself:

> The IIBA is an independent, non-profit, international professional association that is dedicated to advancing the professionalism of its members as well as the business analysis profession itself. IIBA recognizes the important contributions business analysts make to organizations every day . . . The IIBA is seeking to establish common standards of knowledge within the BA profession and is committed to work with practitioners around the globe to continually add to those standards through education, research, and the sharing of effective tools and techniques.[3]

The IIBA was formed in October 2003 with 28 founding members. In April 2006, it became incorporated federally as a nonprofit association under the Canada Corporations Act with headquarters in Toronto, Canada. The organization has grown to more than 18,000 members (as of 2011) and over 100 chapters worldwide.

The Australian Business Analysis Association (ABAA) describes itself:

> The ABAA was formed in 2003 to define, promote and support Business Analysis as a profession nationwide. 'Business Analysis' as a term, provides a collective umbrella for professionals working in the areas of 'Commercial, Process, Technical and Systems Analysis.' ABAA seeks to provide a professional framework supporting those who work in this area.

> The Australian Business Analysis Association is a nonprofit, vendor independent, professional association with the objective to:

> - Define the profession of Business Analysis,
> - Promote the profession and increase public awareness of Business Analysis,

- ■ Develop a Business Analysis competency framework,
- ■ Improve the practice of Business Analysis and the knowledge, competence and standing of its practitioners,
- ■ Represent the profession nationally and internationally,
- ■ Provide a forum for the free exchange of information and ideas.[4]

"Are there any business analyst best practices?"

The Guide to the Business Analysis Body of Knowledge (BABOK), the current version of which is 2.0 released on March 31, 2009, is:

> The collection of knowledge within the profession of Business Analysis and reflects current generally accepted practices. As with other professions, the body of knowledge is defined and enhanced by the Business Analysis professionals who apply it in their daily work role.[5]

The BABOK is similar to what is perhaps the most important document in the project management profession: *The Guide to the Project Management Body of Knowledge* (PMBOK) compiled by the Project Management Institute (PMI). The PMBOK:

> Offers a set of processes, generally recognized as good practice, which delivers results across industries and organizations.[6]

The BABOK describes its mission this way:

> The primary purpose of the BABOK Guide is to define the profession of business analysis. It serves as a baseline that practitioners can agree upon in order to discuss the work they do and to ensure that they have the skills they need to effectively perform the role, and defines the skills and knowledge that people who work with and employ business analysts should expect a skilled practitioner to demonstrate. It is a framework that describes the business analysis tasks that must be performed in order to understand how a solution will deliver value to the sponsoring organization.[7]

The BABOK divides the activities of a business analyst into five knowledge areas. Knowledge areas "define what a practitioner of business analysis needs to understand and the tasks a practitioner must be able to perform."[8]
The BABOK defines the knowledge areas this way:

1. **Enterprise analysis.** "Describes how business analysts identify a business need, refine and clarify the definition of that need, and define a

solution scope that can feasibly be implemented by the business." The knowledge area includes problem definition, developing and justifying the business case, and defining product scope."

2. **Business analysis planning and monitoring.** Concerned with determining "which activities are necessary in order to complete a business analysis effort": It covers stakeholder identification, selection of business analysis techniques, defining a process for managing requirements, and assessment of progress.

3. **Elicitation.** Addresses working with stakeholders to determine what their needs are and "the environment in which they work," and ensuring that we have correctly and completely understood them.

4. **Requirements analysis.** Progressively elaborating the solution definition to enable the solution team to "implement a solution that will meet the needs of the sponsoring organization and stakeholders." Within this knowledge area, the business analyst analyzes the stakeholder information within the current state of the business to identify and recommend improvements and solutions. The knowledge area also addresses validation and verification of the solution.

5. **Requirements management and communication.** Describes the techniques for managing conflict, issues, and changes and ensuring that stakeholders and the solution team are in agreement on the solution. This knowledge area also is concerned with "how knowledge gained by the business analyst is maintained for future use."

   ▪ **Solution assessment and validation.** Covers the role of business analysis once the solution team proposes a solution: assessing proposed solutions, "identifying gaps and shortcomings in solutions," and assessing "deployed solutions to see how well they met the original need."[9]

The BABOK goes on to describe underlying competencies that describe the "behaviors, knowledge, and other characteristics that support effective performance of business analysis."[10] These include analytical thinking and problem solving, behavior characteristics, business knowledge, communication skills, interaction skills, and software applications.

The Question of Certification

"Is it necessary to get certified as a professional business analyst?"

Business analysts in the field seem to be split on the subject of certification. Many already have other certifications. Some, because they are accidental business analysts, do not consider business analysis as their lifelong

profession and are not interested in getting certified. Others, new to the field, are seeking a certification route to establish their credentials in the business analyst world to enhance their careers.

Many business analysts have earned a number of other certifications, mostly from their previous life before becoming a business analyst. A large number of business analysts have the Project Management Professional (PMP) certification from the PMI. This certification may reflect the collateral duties business analysts perform and also that some project managers choose to move in the direction of business analysis even after they have achieved the PMP status.

The IIBA offers the CBAP (Certified Business Analyst Professional) specifically for the business analyst professional. There are also commercial training companies that offer certifications for passing exams after taking their courses, and colleges and universities, such as Boston University, University of California–Irvine, and Villanova have certification programs. There are a number of degree programs for business analysts mostly located in the United Kingdom. Current working business analysts tend to have computer-related degrees, management degrees, or MBAs.

The British Computer Society (BCS) has a certification program called the Information Systems Examination Board (ISEB) that includes a certification in business analysis. The ISEB and the BCS are well respected in the United Kingdom and in Europe. The certification consists of three levels: foundation, practitioner, and higher levels. To gain a diploma in business analysis a candidate must pass two core examinations and two other examinations in selected topics. The ISEB allows candidates who have received a CBAP from the IIBA exemption from the requirements engineering core examination.

The ABAA offers the qualified business analyst practitioner (QBAP) certification:

> The qualified business analysis practitioner (QBAP) is a base-level certification readily available to practicing BAs based on an initial audit of their competencies and experience. This accreditation represents a base level of competency for a business analyst and confirms the bona fides of the skills, training and experience a business analyst asserts at the time of registration.[11]

The Challenge of Business Analyst Certification

"Is there really a way of assessing a business analyst's competence and experience? How would they determine my skills in health care business analysis? It has got to be different than someone in finance or manufacturing."

Because of the breadth of knowledge and skills that a business analyst is required to possess, many of which cannot be determined with a written exam, the concept of certifying a business analyst as a competent professional may be beyond any organization's capability. Consider that it is universally agreed that communication skills are central to the success of the business analyst. How can you assess the professionalism of a business analyst's communication abilities with a multiple-choice exam?

Most certification programs include a number of years experience along with a requirement to pass the exam for requisite knowledge of the profession. None of the programs evaluate that experience to determine whether the experience proves that the professional has been successful at any point. A business analyst who has continually failed to grasp the responsibilities of the role and as a result moves from job to job and failed project to failed project can certainly amass enough time-in-grade to qualify for the certification. Studying for exam and memorizing a manual or taking a class might get this business analyst past the knowledge hurdle. If the business analyst is still unable to communicate successfully or analyze and produce successful solutions to business problems, how is it possible to grant such a person a certification that bestows the prestige of professionalism on them?

It is not a matter of rejecting certification as a means of separating levels of quality in the ranks of business analysts. The real question is whether the current means of certifying really addresses the qualities needed by a business analyst to be successful, much less labeled as professional.

The Value of Certification

"Is the CBAP certification really worth getting?"

That said, the IIBA certified business analyst professional (CBAP) certification is becoming the standard for assessing the basic knowledge of a business analyst. As of the middle of 2011 there are over 1,000 recipients of the CBAP from over 20 countries.

The IIBA lists the following benefits of acquiring the CBAP certification:

- Demonstrated knowledge of the skills necessary to be an effective business analyst.
- A proven level of competence in the principles and practices of business analysis.
- Participation in a recognized professional group.
- Recognition of professional competence by professional peers and management.
- Advanced career potential due to recognition as a professional business analysis practitioner.[12]

The certification requires a level of experience and passing an exam based on the current official version of the BABOK. The first exam was given in Orlando, Florida, on November 10, 2006. The exam is now available in a computerized version at various test centers so that applicants no longer have to sit for the exam in specific locations at specific times.

The CBAP establishes a level playing field for all professional business analysts, organizations hiring business analysts, and organizations considering the establishment of a business analyst function in their organization. The certification provides proof of a basic level of knowledge and understanding of the precepts of business analysis that are essential to the successful execution of the business analyst role.

Knowing the history of the business analyst and seeing that the history of the profession basically mirrors the history of business, what does that mean to you? Well, for one thing it means that the noble profession is not an afterthought of the onslaught of computer technology in business. The business analyst solves business problems, regardless of whether the problem or solution involves information technology. Now that we know where we come from, the next question might be where are we now? Most business analysts appear to work for IT, but is that the right place to be for a profession whose mission is in the business realm? In the next chapter we explore the alternatives.

Notes

1. Adam Smith, *An Inquiry into the Nature and Cause of the Wealth of Nations.*
2. Modern Analyst Media LLC. Available from www.modernanalyst.com.
3. International Institute of Business Analysis. Available from www.theiiba.org.
4. www.aaba.org.au.
5. International Institute of Business Analysis. Available from www.theiiba.org.
6. Project Management Institute. Available from www.pmi.org.
7. International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge,* version 2.0 (March 31, 2009) 2
8. Ibid., 6.
9. Ibid., 9.
10. Ibid., 9.
11. www.abaa.org.au.
12. International Institute of Business Analysis. Available from www.theiiba.org.

A Sense of Where You Are

Strive to make proposed solutions as self-executing as possible. As the degree of discretion increases, so too does bureaucracy, delay, and expense.

—Donald Rumsfeld, Secretary of Defense

From a vice president in the PMO of a large New York bank: "Most of our staff are senior project managers with their PMPs. I was thinking of adding some of our senior business analysts to the staff to fill in spaces that the project managers don't have. Does that make sense, or should I leave well enough alone?"

So, where do you hail from? Today, business analysts are mostly draftees or volunteers; that is, they did not start out in professional life to be business analysts. They were not members of the Future Business Analysts of America club in secondary school or majors in business analysis in college. As more colleges and universities create business analysis curricula, the next generation of business analysts may well be professionals who have chosen business analysis as their life's work. Today, however, business analysts generally started as systems analysts, product managers, project managers, architects, or any number of other professions and ended up as business analysts by either showing a predilection for the role, at least as perceived by management, or were the only ones available when the position was announced. So, are you a former technologist or did you work in some capacity for a business before becoming a business analyst? There is no evidence that a business analyst's previous life has any affect on whether a person is successful in the role of business analyst.

Business Analysts Coming from IT

CIO of a Midwest manufacturing firm: "We're new to the business analyst position, just creating it now. I think we are going to assign the business analysts to work for IT. I'm thinking that each business analyst will be assigned to a project manager. Is that a good way to go?"

The majority of business analysts today gravitated to their position from the technical side of the organization. Most were drafted. Some simply had their job title changed from requirements analyst to business analyst with no change in duties. Others left the technical field for a more hands-on role with the customer or stakeholder.

There are positive aspects for having a technically oriented business analyst. The business analyst with a technical background:

- Has a better understanding of computers and technology and can filter out the operational errors from real problems.
- Can provide quick fixes to the business in emergency situations.
- Generally knows what technology is available to solve problems.
- Can be an extra technical resource on the solution team and can relate to the rest of the team.
- May bring a different perspective or objectivity about the business problem than a business analyst from the business side.

"I transitioned from system analyst to business analyst. Will my technical background help me or hurt me?"

The primary drawback with business analysts coming from the IT side is that they do not have independence. Although their job is to define an IT solution to a business problem, they are constrained to define that solution within the framework of the current IT structure and are not free to discuss the business problem with real platform and solution independence. They are usually in the role of IT emissary, counseling the business as to what IT has available to solve the problem.

When the business analyst has an IT background they:

- Will tend to view all business problems from the perspective of how computer technology can be applied rather than how they can be solved. Many business analysts think of themselves as IT emissaries to the business. They are paid by IT so it is a logical assumption.
- Carries with them the baggage of the ongoing historical distrust of the business for IT and vice versa. The business will naturally view the business analyst as one of them rather than someone to independently

TABLE 3.1 Benefit and Concerns of a Business Analyst Coming from IT

Benefits	Concerns
Independent from the influences of business management.	Conflict of interest trying to represent business to the IT project.
May understand the technical impacts and induce the business impacts to a given solution.	Influenced to come up with an IT solution rather than a business solution.
Has a good relationship with the technicians on the solution team.	Looks ahead at the solution instead of studying the problem.
Is a better filter to discern real problems from the noise due to lack of training or technological unfamiliarity.	Does not have that feel for business. The technician may not really care about the business as much as about the technology.
Can be an additional resource on the solution team.	Keeps solution within the confines of existing systems and knowledge.
Will be able to diagnose technical issues faster.	Jumps to solutions.

assess the problem and produce a satisfactory solution. This makes elicitation that much harder.

■ Will exhibit a tendency to circumscribe solutions within the framework of existing systems and facilities rather than explore unique business solutions that may increase the value of the organizational unit, IT, and the organization as a whole.

This leads to another issue: lack of understanding of the real problem. Since many IT people suffer from Gerald Weinberg's NPS (No Problem Syndrome), it is natural for IT people to arrive at an early solution before the problem is fully defined, implement the solution, and then spend years adjusting the resulting product until it finally solves the original unstated problem. A business analyst from IT is subject to that form of groupthink.

Table 3.1 summarizes the benefits and concerns of a business analyst hailing from IT.

Business Analysts Coming from the Business Community

A director of business relations at a large Pennsylvania bank: "Our business analysts primarily come from the business side. I'd like to keep their affiliation with the businesspeople with which they have relationships. I'm thinking of keeping

the current business analysts with their business units and creating new business
analysts for all the other business units.''

There is much to be said for business analysts with solid business expe-
rience working directly for the business area rather than for IT. The business
analyst is able to build strong relationships with process workers and others
in the business unit. This enables the business analyst to get a deep under-
standing of the problem domain. The business analyst is more likely to sug-
gest non-IT solutions to business problems, and be more concerned with all
the nonautomated activities, such as forms redesign, job description
changes, training, and so on that accompany a typical business process. The
transition from the current process to the changed process is typically
smoother.

As with IT-side business analysts, there are also concerns with the
business-side business analysts. There are technical considerations for
practically all business processes in an organization today. The business
analyst from the business side may not be aware of the technical considera-
tions that may impede their solution, and may not know of technological
advances that may bring about a better solution.

When the business analyst reports to a single business entity within the
organization, he or she will likely be unduly influenced by that business
entity, and especially the head of that entity. The business analyst will most
likely not perform due diligence in verifying alignment with overall organi-
zational strategy. They may not verify that the problem should be solved in
the first place. The boss says to get the thing done, so the business analyst
does.

The business analyst who is so aligned with the business unit is influ-
enced by the processes currently in place in that business unit and does not
explore new and original solutions that may help move the unit and the
overall business forward.

The most damaging issue is that the business analyst will not be moti-
vated to do a full impact analysis of the solution with other departments and
business units. A specific unit pays him or her and therefore their concern
and loyalty are to that unit and its goals. The problem they solve is for that
unit only. The true business analyst must evaluate the solution in light of its
impact on the whole organization and assist upper-level management to
make a decision whether to solve the problem and, if so, what else needs to
be done to enable the solution for all business units affected.

Table 3.2 summarizes the benefits and concerns of a business analyst
coming from the business community.

So just where should a business analyst reside on an organizational
chart—business or IT?

TABLE 3.2 Benefits and Concerns of a Business Analyst Coming from Business Community

Benefits	Concerns
The business area gets a trusted resource.	Too influenced by single business entity.
Full and in-depth knowledge of business area.	Not enough breadth to determine full impacts outside the business area.
More in-depth knowledge of systems supporting business area.	Does not consider available technology in the solution because of lack of knowledge.
Can act as or be considered a subject matter expert, thus relieving the business area from identifying and assigning subject matter experts to a project.	Solutions circumscribed by the way things are done now instead of evaluating new potential solutions.
More likely to identify a non-IT solution where applicable.	Accepts symptoms as problems instead of investigating thoroughly.
Aid in the transition from old process to changed process.	Becomes regarded as one of them by IT allies.
Reduce risk by working within the confines of the business.	Tend to see changes as only incremental improvements.

Living with the Business

"We work for IT and we are assigned to a business unit. Sometimes I hardly ever talk to someone in IT except for reviews. Is this the way it is done everywhere else in the industry?"

One solution is to co-locate the technically oriented business analyst with the business unit that person is supporting. This co-location promotes a better relationship between the business and the business analyst, and psychologically the business analyst feels as though they are representing the business since they have to travel to see the solution team.

For example, Motorola divides IT staff into three teams: Plan is composed of business analysts who interact with the rest of the business, Build is composed of application developers, and Run is the maintenance staff. The Plan and Run teams are housed with the business.

🔍 Example

Issues with the Business Analyst Co-Located in the Business

I was brought into a large petroleum company in Houston in the early 1990s to assist the IT department in a transition from a centralized to a distributed model. They were moving their business system analysts out of IT and assigning them to work full time for the business departments.

Most of the business system analysts had backgrounds as programmers and many worked as one-man bands, defining requirements, designing, coding, and testing the solution. The business system analysts anticipated that they would be doing the same type of work, except directly for the business. Their primary concern was not the change in management; it was their standing in the IT community. They felt that they would be stereotyped as accounts receivable business analysts and end up only working on A/R systems, which would decrease their value as general technicians.

The company felt that having the business analysts work directly for the business, being charged to the business budget rather than the IT budget, would reduce the overall operating expenses for IT, considered at the time a cost center. Moreover, upper-level management felt that by assigning the technical personnel directly to the business units the problems would be solved faster and better and the units would have more control over the results.

The company reverted to a centralized IT department after about five years of the distributed model. They discovered in practice that:

- Many technicians assigned to the business units left to join other IT organizations where they could work with their peers.
- There were some business units that still found that they could not get all the planned problems solved with the technical personnel the unit was assigned.
- There were other business units that had excess capacity and were unable to keep their assigned technicians fully occupied with meaningful development or maintenance.
- The business units were not cooperative in lending excess resources to one another.
- The technicians were not suggesting innovative technical solutions to the business problems, opting instead to simply improve what the business unit already had because the process workers resisted change.

The Lone Ranger

"The BABOK talks about a team of business analysts working on a single project and how you need to plan the work for all business analysts. Does that actually happen? I'm lucky when I am only assigned to one project, and that's rare. Mostly I'm on two or three. And the only time I ever work with another business analyst is on the holiday party."

One of the aspects of business analysts that I find interesting and restricting is the tendency, especially among those who come from a background of programming or engineering, to be a lone ranger.

Until the advent of agile practices such as team ownership of code and pair programming, the programmer worked alone laboring over one program, asking for help occasionally, about as often as the programmer would read instructions or ask for directions. The program was the product of the programmer and, usually, the programmer's ego, status, prestige, and income were wrapped up in that program.

The same holds true for business analysts in many organizations. The business analyst produces a set of requirements that state the solution to the business problem he was assigned to solve. No other business analyst is involved with that solution document. From the organizational perspective this makes responsibility and accountability a lot easier: There is only one person to blame when things go wrong.

There are a number of issues with this approach to business analysis. The issues are pretty much the same as has been addressed with programmers working in agile system development methods. When a single programmer codes the programs or classes, there is a single point of failure in the development process and, to a degree, during production. Moreover, the programmer treats the program as their own work of art: They test less (because after all it should not fail), they cannot take criticism about the program (making it hard to institute inspections or reviews), and when things go wrong, it cannot be their program that is causing it. Sounds like a doting parent and his child, doesn't it? The agile methods of software development have stringent rules about team or joint ownership of code: Any programmer can change any code and is responsible to fix any error in any code they find, whether they wrote it or not. This reduces the single programmer issues.

Business analysts can get the same possessive attitude toward their solution document when they work alone. Unfortunately, where programmers can work together in teams, business analysts usually don't. For most medium to small projects in organizations, the business analyst is lucky to be able to devote all his time on defining the solution and not also have to wear other hats.

Most organizations today assign single business analysts to multiple projects, especially where the projects are in the same business area. The rationale is that the business analyst is only on the project for the length of time necessary to define the requirements and when multiple projects are being run in a specific constituency in the business, the business analyst can define requirements for these multiple projects at the same time. On larger projects there may be a single business analyst assigned solely to that effort. And of course when that happens, there is a single point of failure in the software development process, especially when the process is linear in nature and cannot proceed until the business analyst has produced the approved requirements.

To many, perhaps most, business analysts, this is an acceptable approach. Most problems are simply not of the size or scope to warrant a team of business analysts swarming over two or three process workers. Interaction among business analysts in these organizations is minimal and mostly accidental: A business analyst needs information from an ongoing project and finds another business analyst assigned to that project a more conducive source of information than the project manager.

Working Both Sides of the Street

"We will start out with 20 business analysts and we want the group evenly distributed between those recruited from the business and those from IS. We also want all 20 to be at the same level of expertise when the business analyst organization kicks off."

Many organizations are starting to purposely populate their business analyst teams from both the business and technological communities. The mix varies. A California insurance company purposefully decided to populate their new 20-person BA team with 10 people recruited from IT and 10 from the business. In a financial support company in New York, the business analyst manager reports that the percentage is 70–30 in favor of business analysts from the business community. In a government consulting company in Virginia, the ratio is in favor of the technological team by a 60–40 margin.

The business analyst manager in a New York–based support services firm for global financial institutions describes what can be considered a typical business analyst organization: "I have 12 in New York and 8 in Hyderabad, but these are just in my business unit, there are about 100 firmwide. . . . Most of our BAs are college graduates with either a business (usually operations) background or a development background. For business backgrounds, we are generally looking for around 10 years of experience. For development backgrounds, around 5 years experience. Generally, pay

is higher for business side analysts." Seventy percent of the company's business analysts are from the business side.

This kind of mix of business and technical perspectives and backgrounds on the overall business analyst team brings positive results, even when the business analysts are working as lone rangers on projects. Assuming the business analysts are co-located, propinquity alone will bring more technical awareness to the business-based business analysts and more business sense to the technical-based business analysts. The approach again borrows from agile development, which uses a concept of generalizing specialists on its multifunctional teams. Each member of the team has a specialty, say programming, but is also expected to know something about the other specialties like testing, database administration, business analysis, and so forth. What is the best way to get the business analysts in the organization to work on both sides of the business-technology street? The next section addresses that question.

Central Business Analyst Organization

It has become clear over the past 10 years or so that some type of formal or informal centralized business analyst group within the organization is a successful implementation model for an organization. In a centralized business analyst approach, the business analyst group operates as a separate unit, not reporting to either IT or a business unit. This approach sets the business analyst department up as an independent entity able to define and solve business problems on behalf of the business as a whole rather than any individual constituent of that business. An independent business analyst function in the organization provides a staff with the experience and insight to identify the problems of the business and identify the correct process solution.

Among the benefits of a centralized organization are:

- Each business analyst gets exposed to multiple business areas regardless of their assignments.
- The business analyst avoids getting stale working in one area because he or she is dispatched to other business areas of the organization for each new project.
- Able to leverage new ideas better through contact with other business analysts.
- May get new ideas and concepts from other business analysts and from other business areas.
- Better able to gauge business impacts of solution proposals because of wider exposure to whole organization.

- Creates and maintains a common business vocabulary for both the business and IT, as well as all the business analysts.
- Centralized training and knowledge base—lessons learned, tricks, tips, and techniques.
- Easier to acquire and use business analyst tools when the business analyst group services the entire organization instead of a single business unit or IT. Business analysts have more cross-functional and cross-domain business experience, rather than just focusing on one area or function within the business.

Although an unaligned, standalone, business analysis unit is best, there are pragmatic, cultural, and political reasons why an independent business analyst unit will not work in many organizations. When the reorganization cannot be officially made, it is possible to move the existing BA staff in the direction of non alignment, perhaps informally.

An informal independent business analyst organization would still have the ability to:

- Document the group's learning and approach in developing business requirements and solutions.
- Conduct postmortems on all requirements development and problem-solving efforts regardless of size, and incorporate the learning into the next project.
- Share impact information about a wider range of the overall organization.
- Create a common glossary of business terminology.
- Create and maintain a standard business analyst process.

Knowing what you are as a business analyst, and where you come from, and even where you are, or should be, is good. That creates a framework within which you can work. It gives you a sense of history and legacy. The next aspect of the business analyst to focus on is quality. What makes a good business analyst? What areas of my personality or skill set should I pay attention to if I want to improve my business analyst performance, or perhaps change careers and become a full-fledged business analyst? The next chapter addresses those skills, knowledge, and personality traits that distinguish the top-notch business analyst.

What Makes a Good Business Analyst?

Experience is that marvelous thing that enables you to recognize a mistake when you make it again.

—F. P. Jones

A successful process defines the activities and practices for success so that anyone can follow the process and achieve a successful result. The truly successful process also has a dependency on the skills and experience of the person who is executing the process. A carpenter follows a process to build a house, a process that is created heuristically through trial and error, and which can be taught to apprentices. The carpenter produces a house that matches the blueprint, meets the expectations, and does so in an efficient way. I can follow that same process to the letter but, not being skilled or experienced as a carpenter, my house will be significantly less livable and I will probably have a pile of scrap lumber from the mistakes made along the way. Similarly, the business analyst process requires certain skills for success. A person can follow the process as defined in this book and elsewhere and produce a satisfactory product without any of the skills associated with a business analyst, which is the beauty of a process. However, the more skilled the business analyst is, the more likely the product will successfully solve the problem and the easier the process becomes. Let's define the basic skills of a business analyst.

"What is the skill set required for business analyst? What are the key skill sets a business analyst must possess? What is more important: technical knowledge or domain knowledge?"

In my role as writer and consultant to business analyst organizations, I am constantly asked what skills are necessary to be a business analyst. Usually the questions come from technical types who are considering a career change before their job is outsourced overseas. Sometimes the question is from businesspeople who would like to move in a different direction. Other times the question is from a nephew or other relative anxious to find a way to break into a promising career. I am also asked by upper-level management what I would look for to populate their fledgling business analyst organizations.

In this chapter we explore the technical and other skills necessary for being successful as a business analyst. We also look at the traits that help the analysts apply those skills.

The Skillful Business Analyst

My cousin's son asks: "I am graduating from college and I am thinking that business analysis might be a good profession. What do I need to be successful? Actually, what skills do I need to be hired?

The business analyst is not only a combination of skills; the business analyst is a combination of left-brain and right-brain thinking. The business analyst needs the logic and the objectivity of the IT professional and the instinct and human relations sense of the businessperson. Although specific background requirements, skills, and traits vary from one organization to the next, the most common attributes listed include:

- Ability to ask the right kinds of questions and the curiosity to dig further.
- Ability to gather and understand information about the business and business processes.
- Ability to analyze information to determine what is necessary to be done to solve the problem to the detail necessary for the solution team to implement the solution.
- Ability to plan, develop, and conduct acceptance testing of applications and systems to prove that the business problem has been solved.
- Ability to build solid, trustworthy, working relationships with product stakeholders and business management.
- Ability to troubleshoot production application problems quickly and accurately and determine a solution.
- Ability to communicate verbally and in writing using clear, precise language.
- Ability to facilitate and moderate meetings or workshops.
- Ability to identify workflow problems in current systems.

Is a Business Analyst Born or Made?

"I have been a programmer for thirteen years. I think they are going to send all the programming offshore. At least that's the rumor. I don't know what they will do with us. Can I shift over to becoming a business analyst?"

The business analyst is not a technologist with knowledge of the business, nor a businessperson with some special technical knowledge. The business analyst understands both general technology and the specific business area with the problem and has the ability to merge the two successfully to get the best of both worlds.

There are a number of attitudes a business analyst should possess that are not necessarily trainable, but they can be learned or adopted. These outlooks on life are certainly not mandatory; however, having these attitudes contributes to the success of the business analyst.

Accepting Ambiguity

At a financial company in Delaware, Ruth, a senior business analyst said, "I am an expert in detecting ambiguity. I have to be to do my job."

As Ruth notes, the business analyst needs to be aware that nearly everything in business is ambiguous and that there are many sides to every problem and solution. This attitude is important, especially when dealing with technologists who are used to working with computers that are binary and unambiguous. Those business analysts who hail from the technology side of the house may find it more difficult to accept this natural ambiguity. At the same time the business analyst cannot forget that the documented solution must be unambiguous so that the technologists can implement it.

Tolerance for Perturbation

The business analyst accepts the fact that there is turmoil and change. The successful business analyst is able to see change as an ongoing event, a flow, a process. Change is not sporadic and singular. It is not something that is turned off and on as needed. Once a project is finished and the product is delivered to the business and placed into production, the business analyst reviews the results to look for more problems, and continues the process of changing the business for the better. The business analyst recognizes that the current project is just part of a flow of change and improvement to the organization.

Curiosity

Mary, a member of the PMO at a large financial institution in the San Francisco area, said that the primary aspect of business analysts is that they "must

be curious. Even if they aren't interested in a particular subject they should be curious as to why it happens and how it happens. A business analyst needs curiosity." The business analyst is interested in finding out why things happen, why processes are set up the way they are, and why people act and react the way they do in the business environment. The business analyst is also curious to know the results of change.

Impartiality

The business analyst's allegiance is to the organization or business, rather than to either IT or to a specific functional area of the business. Even when business analysts are assigned to be accounts receivable business analysts, or claims processing business analysts, their perspective must be that of the entire organization. It is this objectivity that sets the business analyst apart from everyone else and drives the successful business analyst.

So What Does It Take to Be a Business Analyst?

It takes good communication skills, analytical abilities, perseverance, a fast learning curve, the ability to adapt to continual change and ambiguity, and above all, a healthy sense of humor and a good appreciation for the absurd. Here's why.

Communication

The primary quality or talent a good business analyst must have is the ability to effectively communicate. It is at the center of everything a business analyst does. If you are reticent and fearful of meeting people, especially in a one-on-one situation, and the thought of making a presentation tomorrow reminds you that tomorrow is a sick day, perhaps the job of business analyst might not be your best choice.

Communication skills are where the typical technician stumbles in his attempt to become a business analyst. Without training, a technician may not have the sensitivity to be aware of the real communication, what to say and when to say it, when to communicate in person and when e-mail is appropriate, how to handle bad behavior in a meeting, how to resolve conflict so that everyone is on board with the resolution, and so forth. Fortunately, communication skills can be acquired through training, reading, and practice. Some people will find it much easier to learn to communicate; some will find it very difficult, even painful.

> ### 💡 May I Suggest?
>
> **Looking to Hire . . . ?**
>
> If you are looking to hire a business analyst or are considering a request for position change, how can you tell if a person is a good communicator as a business analyst? Ask the candidates questions that require a description of an event in which they participated or witnessed. When the candidate tells you a story and engages your interest, you have a potential business analyst. When the candidate states only the facts or covers just the highlights, then the candidate will need training in communications to qualify as a good business analyst.

Analysis

The ability to analyze is a close second to communication skills for the business analyst. A business analyst reduces the business problem into logical components to identify the correct solution. The mark of an experienced business analyst is being able to cut through the jargon, excuses, and rationales to get right to the real business problem.

Analytical skills can also be learned. Much analysis consists of diagramming and modeling to get better views of the overall problem domain and potential solutions to the problem. These techniques can be learned through books and classes. As with communication skills, analysis comes easy for some people who are used to questioning and challenging everything. Those who accept things at face value without question may find it difficult to learn to be a business analyst.

Technical and/or Business Knowledge

Many companies are requesting specific business or technical knowledge when recruiting business analysts. Some HR departments or recruiters require such knowledge as a mandatory part of their recruiting process to help reduce the number of applicants. Other companies are hoping to hire the experience along with the additional staff. However, technical knowledge can be learned. It can be picked up by osmosis working with technical people if nothing else. Business knowledge is also learnable. It can be learned on the job as the business analyst investigates the problem and solution. Asking questions, listening carefully, resolving misunderstandings, and reducing assumptions are all necessary ingredients to quickly learn about new and changing technology and business environments.

We can see that a business analyst is a multitalented, general purpose, problem solver for the organization and that business analysis has a long and illustrious history filled with luminaries of great repute. A business analyst can be proud. In Chapter 5 we turn our sights to what a business analyst does, or at least is supposed to do, before we address how the business analyst actually does all the things expected of him.

Roles of the Business Analyst

It is probably no mere historical accident that the word person, in its first meaning, is a mask. It is rather a recognition of the fact that everyone is always and everywhere, more or less consciously, playing a role.

—Robert Ezra Park

When it comes to solving problems in the business, the business analyst takes on a number of roles. As previously noted, there are a variety of skills needed to be a successful business analyst and these skills are needed to play these many and varied roles. Let's now identify these roles and where they are played in the overall process of bringing value to the organization.

💡 May I Suggest?

Most everyone in business, especially in tough economic times, ends up playing multiple roles. The business analyst, too, plays a number of roles as described in this chapter. To do so effectively, you must be aware of what role you are in and make sure the responder knows which role you are playing during the communication. It's also best to not mix up roles within a single communication.

During the lifetime of the project, or product, or for that matter, the lifetime of the business analyst, the BA plays a number of different roles for

the organization. Though some roles will be comfortable and others not so comfortable, all roles are played.

I routinely ask business analysts in an organization what they do, what is their role. The answers I get span the scope of the entire IT spectrum except for actually writing the code.

Here is a short list of the primary activities that business analysts do as part of their day-to-day job. This is not what management thinks, or what HR puts on the job description. This list is distilled from the responses of hundreds of business analysts over the past seven years who are actually doing these tasks.

- Define business problems
- Manage release schedules
- Determine what broke after solution is installed
- Communicate to key stakeholders
- Organize and run meetings
- Train users, and so on
- Define procedures for business
- Write requirements
- Write the business case
- Write test plans
- Document everything
- Work to improve customer satisfaction
- Educate users, other teams, management
- Analyze trends
- Act as an SME (subject matter expert)
- Interpret what the user is looking for
- Understand business procedures
- Provide guidance to stakeholders for devising effective and efficient approaches to achieve the project objectives
- Identify and resolve issues
- Manage the risks
- Liaise with other project areas to coordinate interdependencies and resolve issues
- Liaise with various business units to gather requirements and resolve issues
- Improve business processes
- Analyze and map processes (current state/future state)
- Analyze data
- Produce high-quality documentation
- Report status and issues to the project manager(s)
- Contribute to enterprise architecture development from a business needs point of view
- Test the modification and new systems
- Create project plans on behalf of the project manager
- Create or establish business process—define what the users do
- Mediate all the different user's usages
- Determine what is working and what is not

- React to change in scope or direction
- Find holes in requirements
- Understand customer needs
- Create use cases
- Review service requests
- Maintain the traceability matrix
- Write users' manuals
- Act as a liaison
- Resolve conflicts
- Prepare RFPs (requests for proposal)
- Evaluate RFPs
- Run benchmarks
- Perform risk analysis
- Liaise between users and developers
- Work with the users to find out what they want
- Document system for the developers
- Make sure what is built is what the users want
- Communicate and coordinate
- Attend meetings
- Facilitate and coordinate meetings
- Put out fires on installed systems
- Explain the language of the problems
- Create documentation that the user and developer will understand
- Provide ongoing support for users
- Be the initial contact for the users
- Do a whole lot of documentation
- Coordinate testing effort
- Execute system testing (defined as ensuring the product matches what the business analyst specified in the requirements document)
- Estimate project work
- Manage change
- Manage the demand of incoming requests

Why such a diverse list of activities? A manager might explain that the business analysts simply do not know what they are doing. The business analysts might counter that the organization has not really defined the role of a business analyst. The list actually demonstrates the breadth of the business analyst's reach in the organization, and the importance of the role.

The business analyst plays each of the roles listed in this section on a continuing basis. Each role has a specific goal to be achieved in the overall process of solving the business problem and adding value to the organization. Each role requires individual focus when it is being played, much like an actor taking on multiple roles on stage or in a movie.

The roles in this section are those that are typically listed in job postings for business analysts or in business analyst job descriptions. Whether or not the roles are listed, usually the organization expects the business analyst to play these roles. There are undoubtedly many other

roles a business analyst plays in various organizations, which are ancillary to these.

> *Good communications are as stimulating as black coffee and just as hard to sleep after.*
>
> —Anne Morrow Lindberg

Communication is the essential ingredient in every role a business analyst plays; whether you are eliciting the information as an investigator, explaining or translating jargon to divergent communities, or defining the solution to the problem, it all involves communication. The successful business analyst does not limit their communications to just the path between the IT and business communities. They communicate throughout the organization: among business units, from upper- to lower-level management, to entities outside the organization including vendors, regulatory agencies, and customers. And so we see the first role of the business analyst is as an intermediary serving as the interface between and among the various constituencies involved with solving the business problem. Later we discuss the roles of diplomat, change agent, and problem solver, among others. Each has its own principles, rules, and guidelines and each is anchored in communication.

Here are the primary roles of the business analyst:

- ■ *Intermediary*. This is the most common role the business analyst plays, functioning as a go-between among the business, IT, and upper-level management.
- ■ *Filter*. The business analyst plays the filter role when receiving and evaluating change requests from the business whether for a prior bug fix or a completely new system.
- ■ *Investigator*. This is the role played when eliciting information to determine the problem and solution.
- ■ *Facilitator*. The business analyst is often a facilitator, helping the business, upper-level management, and the solution team understand problems and work together to solve them.
- ■ *Diplomat*. In this role, the analyst assists in the resolution of conflict among the parties and negotiates collaborative solutions.
- ■ *Business Process Improver*. The business analyst looks for ways to improve business processes.
- ■ *Quality Assurance*. In the role of quality assurance, the business analyst makes sure the solution solves the problem completely and effectively.
- ■ *Change Agent*. The business analyst adopts the role of change agent to make sure the solution is accepted by stakeholders throughout the

development process and then efficiently placed into production and used to generate value for the organization.

Intermediary

The analyst is the middleman between the user, who decides what has to be done, and the development team, which does it.
—Tom DeMarco

As defined by Dictionary.com, an intermediary is "an intermediate agent or agency; a go-between or mediator" or "acting between persons, parties, etc." The business analyst acts as an intermediary between the business community and the IT community to assist in communications between the two. This is considered by many to be the primary role of the business analyst. However, business analysts themselves are finding that being "in between" the business and IT is not a savory position. With the ongoing movement to bring technology and business closer together, sitting at the same table and discussing issues directly, business analysts who fancy themselves as simply intermediaries are finding themselves squeezed out of jobs. In truth, we business analysts find that running interference for either IT or the business is a thankless job and takes valuable time away from solving business problems.

In the end, though, there is no way to avoid being in the middle. It is part of our job and responsibility. We can, however, make sure we understand the role of intermediary we are playing.

"How can I improve the communications between stakeholders and business and developers?"

There are several variations of intermediary you may adopt. You may have to adopt one or more of these intermediary roles, sometimes for the length of the project. Sometimes you may alter your stance during the project as conditions warrant. Sometimes you appear to be one kind of intermediary to one constituency and another kind of intermediary to another.

Here are some stances the intermediary may take, presented in order of activity from less active to more active and involved, thereby requiring more expertise to carry off successfully, and carrying with it more risk.

Bridge

Organizational management, both on the business side and in IT appear to be clinging to the view that the business analyst is simply the bridge

between the technical people who run IT and the businesspeople who run the business. This view of the business analyst has developed over the years.

A bridge is passive. Traffic flows across the bridge. The bridge itself performs no activity, analysis, or evaluation. When you assume the role of bridge you become nothing more than a conduit and you are also liable for all communication failures, just like a bridge that collapses and interrupts traffic. This is not the best stance for the business analyst to take.

An example of the business analyst as a bridge might be seen in the following scenario:

> Don Foley, director of finance, calls Bob Anderson, the business an-
> alyst, into his office. "Bob, I need you to take these requirements
> over to IT. Let them know what we need. I hate to talk to those
> nerds."

> Paula Morgan, the project manager, meets with Bob and reads
> Foley's list of requirements. Paula replies to Bob after reading the
> list: "Bob, tell Don that we cannot do everything on the list for the
> budget that we have been allocated. He's going to have to cut
> something out. Those people are always trying to get more out of
> us than they pay for."

> Bob returns to Don with the message.

When the business analyst acts solely as a bridge there are a number of negative repercussions:

- An increase in overhead to solve the problem.
- An increase in risk of miscommunications or mistranslations.
- An increase in time to go from sender to receiver.
- An increase in distance between the problem and solution.
- A continuance and perhaps furtherance of the separation of IT and business.

Clearly, none of these benefit the BA or the organization. To show how the definition and role of the business analyst is changing, the *Business Analyst Body of Knowledge* (BABOK), version 1.6, defined the business analyst as, "a key facilitator within an organization, acting as a *bridge* [emphasis mine] between the client, stakeholders and the solution team."

In the subsequent version, 2.0, released in late 2008, there is no mention of the term bridge in the definition of the business analyst.

Liaison

As defined by Merriam-Webster online, a liaison provides "communication for establishing and maintaining mutual understanding and cooperation."

The liaison still refers to the middleman between IT and the business, but this definition clearly indicates that a liaison is more active than a bridge or conduit.

According to the BABOK, business analysis is:

> The set of tasks and techniques used to work as a liaison among stakeholders in order to understand the structure, policies, and operations of an organization and recommend solutions that enable the organization to achieve its goals.[1]

Revisiting our scenario: Bob, the business analyst, now acts as a liaison. Bob takes a more active role in the communication between Don and Paula. The scenario now plays like this:

> Don Foley, director of finance, calls Bob Anderson, the business analyst, into his office. "Bob, I need you to take these requirements over to IT. Can you look at the list of accounts payable requirements on pages three and four? We might need to change some of them."
>
> "The project team may find it difficult to do all these with the budget they've been given."
>
> "That's true. See what you can do."
>
> Paula Morgan, the project manager, meets with Bob and reads Foley's list of requirements with Bob's modifications. As Paula reads the list, she states, "I know you've told Don we cannot do everything. What are his high priorities?"
>
> Bob explains which requirements should be done first and Paula agrees. Bob takes the message back to Don.

As a liaison you make things happen and take an active part in the communications between IT and the business. You may also function as liaison among several business units participating in the problem definition or the solution. In all cases your job is to be in the middle and not take sides.

Translator

A business analyst understands both technical jargon and business-ese and is often called upon to translate the two languages.

The technological terminology of the solution team is foreign to the typical process worker and the jargon of business is unfathomable to the computer technician. While there are those, especially in the agile community, who agitate for each community to become conversant with the other's lingo so that they can discuss the issues without an intermediary, this does not appear to be the path many are opting to take. The general consensus is that it is easier to have someone in the middle translating the form, content, intent, and language, preferably in short hand, from one community to the other.

Here is an excerpt from the *Business Analyst's Handbook* (from a public utilities company) compiled by a highly effective IT organization on the East Coast of the United States:

> The business analyst receives information or artifacts from the Account Manager, project manager, and client sponsors; they *translate* [emphasis mine] these into deliverables that the client sponsors, systems analysts, application architect, data architect, and end users can act upon.

In many cases, more than the language has to be translated; cultures need translating as well. The developers expect communication from the business to be, "Tell me exactly what you want and I'll build it that way." The business says, "I'm not sure what the solution is, but I'll know the problem is solved when I see it." The IT solution team works with precision and unambiguous bits; the business deals in approximations and anomalies. The designers and developers want an elegant solution; the business wants it only good enough to turn a profit.

Which brings us to another concern about business analysts acting solely as translators: The sender of the message expects the translator to translate not just the words uttered, but also the intent behind the words. And this is a problem. The problem is exacerbated in the business setting where sometimes the translator is used primarily to send unsavory information to the other side.

Let us look in on our scenario with Bob acting as a translator:

> Don Foley, director of finance, calls Bob Anderson, the business analyst, into a meeting with Paula Morgan, the IT project manager, to discuss the new accounting project. Previously Bob had explained to Paula what all the requirements on Don's list mean.

> After reviewing the requirements briefly, Paula says, "To get this done within our timeframe we'll need to timebox it, perhaps using XP and we'll need an onsite customer."

Bob explains to Don, "Paula is thinking of using Extreme Programming, an agile software development methodology which uses short, fixed-duration iterations to develop the software with a lot of feedback from your people. In fact, to make it successful, she would like one of your mid- to senior-level people to join her project team full-time for the length of the iteration which is usually about two weeks." He adds, "They are pretty sure they can get all the requirements done using this method."

Be careful of your translation when acting as a translator. Check with the sender after translation to make sure you got it right, assuming the sender understands what you said. Also be careful to translate just the language and jargon, and not interpret the intent.

Emissary

The business analyst often acts as an emissary when performing the intermediary role. According to Dictionary.com, an emissary is a "representative sent on a mission or errand or to represent the interests of someone" or "an agent employed to advance, in a covert manner, the interests of his employers."

An emissary represents one party to convince the other party of some course of action beneficial to the first party. The emissary is granted enough authority to negotiate with the other side to achieve certain objectives for them. A senior analyst at IBM related that he has heard business analysts referred to as customer proxies who "answer designer's questions about the application area, and render decisions on behalf of the customer."

Many business analysts view their role entirely as emissary or stakeholder representative and not at all as a member of the solution team. In that role, the BA has these responsibilities:

- Faithfully represent the views and needs of the section of the broader stakeholder community they represent.
- Take an active role in the project.
- Participate in requirements and other project reviews.
- Participate in the assessment and verification of the product produced.
- Attend workshops and meetings.
- Do independent research.
- Champion the project to the stakeholder they represent.

In our ever-changing scenario with Bob, Don, and Paula, Bob is now an emissary for Don:

> Don Foley, director of finance, calls Bob Anderson, the business analyst, into his office. "Bob, I need you to take these requirements over to IT. Now I know they will resist this long list, so I want you to work with them to whittle it down. I cannot add anything to the budget. All the features listed on pages four and five are low priority but try to get as many as you can into development for this release. Don't give up anything on pages one and two."

> When Bob meets with Paula, he says, "I know you are going to push back on this list, Paula. I have to tell you that Don does not have any more budget, so the compromise is going to have to come in some other way."

> Paula assumes she is talking to Don through Bob and that Bob has the authority to make decisions about the project.

When assuming the emissary role the issue is that the receivers assume you are aligned with the senders and are no longer taking the objective stance of the business analyst. Once you have represented the business community as an emissary, the solution team will pigeonhole you as one of them and you will not be able to stay neutral.

On the other hand, the business analyst cannot simply be an emissary from IT to the business because the best solution to the business problem may not be an IT solution. "Solutions often include a systems development component, but may also consist of process improvement or organizational change."[2] An IT emissary necessarily looks for IT solutions to all problems.

Many times an emissary ends up in the game of shuttle diplomacy between the two parties. First the business analyst acts as emissary from the business to the solution team and then returns as emissary from the project manager to the business, and so on.

Referee

The referee is an intermediary between two teams vying for the same or different goals and both teams have agreed to a set of rules that govern the game. Note that the referee's role is only played when both sides know there are rules, whether they know what the actual rules are or not, and both sides acknowledge the role of referee. The referee is totally impartial. The referee is a rule enforcer and occasional rule arbitrator, not a rule maker.

It is a rare circumstance for the business analyst to act as a referee by long distance. Usually refereeing takes place in a meeting. Here is another variation on our scenario to illustrate:

> Bob Anderson, the business analyst, calls a meeting between Don Foley, director of finance, and Paula Morgan, the IT project manager, to discuss changes to the new accounting project.
>
> Don starts before everyone has taken their seats. "Listen, there is no way we can accept the system without these changes."
>
> Bob says, "Don, I think it would be better if we were all sitting down first."
>
> Paula responds, "To put those changes in, we will have to add three weeks to the schedule."
>
> Bob says, "We can't do that. The corporate policy requires us to have the system up by the date we've agreed to."
>
> Paula is not happy. "This is *not* going to happen with this schedule," she says with emphasis, jabbing her finger at the Gantt chart on the table. "And we're not working overtime to make it happen."
>
> Bob says, "Okay, you are right, Paula, we're not doing overtime because HR has stopped overtime pay for the rest of the fiscal year. Before we go on, I think we should step back five yards here and take another look at the requirements and see if there is something we can do there. That is what we are here to discuss."

Note that, unlike a mediator, the referee does not suggest solutions or options. The referee only keeps the rules of engagement and enforces them in the game.

The clear difficulty when you adopt the referee role is that sooner or later both sides are convinced you are wrong, arbitrary, vengeful, blind, unfeeling, and probably paid off by the other team; in other words, all the feelings against referees that fans and teams have in organized and disorganized sports.

While the normal referee role is played between business and the solution team as indicated in our scenario, at a large package delivery company the business analyst for a project team regularly plays referee between the developers and the quality assurance team on questions of defect identification. The testers identify defects in systems under development and the developers protest that the identified defects are really enhancements, or not in the requirements, or not defects at all. The business analyst is called upon to mediate or act as referee to resolve the situation.

Business to Business

While the common understanding of the liaison role is between the business and IT, in many cases business analysts find themselves liaisons between various business constituencies who have some interest in the change being made to the organization. Similarly, the business analyst may be a translator interpreting the various business terms for the different constituencies involved in an enterprise project. The business analyst may act as an emissary from the requesting constituency to other involved business groups. And the business analyst may indeed be a referee among divergent business constituencies who may be agitating for specific and conflicting functionality in the proposed solution, for instance:

> Bob Anderson, the business analyst, meets with Don Foley, director of finance. "I can't seem to get Sara on board with this program."

> Sara van Pelt is the vice president of sales. "She is unhappy with the change in the way we will be processing commissions. Bob, you know more about the system than anyone. Perhaps you can explain it to her so she will understand why it's good for everyone. Oh, and by the way, could you also meet with Mary, the manager of accounts payable? Her troops seem to be resisting the change and want some new requirements that don't quite match what we want."

It is hard for a business analyst from IT to step in between two conflicting business units. The credentials issue looms large: "You are from IT. What do you know about business?" The business analyst is the natural mediator in the business-to-business conflict because the conflict is about the business analyst's solution and who better than the business analyst to explain it to all sides?

Playing the Intermediary Role

The whole intermediary role can get complicated when the business analyst acts as different types of intermediary in the same meeting, as in this final variation of the scenario:

> Bob Anderson, the business analyst, calls a meeting between Don Foley, director of finance, and Paula Morgan, the IT project manager, to discuss some problems on the new accounting project.

> Bob starts the meeting off by saying, "I brought Don's accounting requirements to Paula earlier (bridge), and she had her systems analysts look at it. Apparently there are some issues we need to iron out. I think that we can get them resolved here" (liaison).

"Well," says Paula, "There is an issue with passwords. The security policy requires a different password for each of the designated secure databases, and you are using five of these databases. Your people will need to enter five passwords."

"That's absurd," hisses Don. "We can barely remember one password without writing it down. We can't have five passwords. The requirements call for only one, and that's what we'll have."

Bob interjects, "That is the rule, Don. It is in the security policy." Bob taps the four-inch thick security policy manual in front of him with his finger for emphasis (referee).

"Well," offers Paula. "Steve Albert, our systems analyst, says that we could construct a shadow and place it on an intermediate server over an SSL connection. By setting a list-link association we may be able to create surrogates that access the database security and open all the tables with a single sign-on. Of course that requires specialized resources."

Don stares at her blankly and then at Bob. Bob explains, "Her team can write software that will allow you to only use one password and still provide the necessary five passwords to stay within the security policy. It will cost more money" (translator).

Paula adds, "We'll need a consultant and two more weeks in the schedule."

Don shrugs. "All right. Bob, you'll have to go talk to the accounts payable people and make sure they are onboard with an extra two weeks" (emissary).

Don makes some notes and then says, "All right, what's next?"

To avoid miscommunication, you need to be clear about the goal of the intermediary and the goal of the parties using the intermediary. Complications arise when IT believes you are a bridge and you assume the role of emissary and the business expects you to act as a referee. Clarify your own role, and communicate clearly to the constituents.

Principles of the Intermediary

"How do I work with the teams—both IT and business?"

There are some general principles that apply to an intermediary regardless of which stance the intermediary takes:

- ***Deliver the message***. Delivering messages is a routine activity of the intermediary, regardless of the role in which you cast yourself. The messages are most likely oral, but may be written formally or informally. Some business analysts have expressed frustration at being what they referred to as message boys. Unfortunately, some ferrying of messages back and forth between IT and the business goes with the territory. Unless and until you can get the business community and solution team comfortable sitting at the table and discussing issues directly, you will always be carrying messages.
- ***Understand the message***. Assume that when you deliver the message there will be questions about its content. Make sure that you fully understand the content and intent of the message before you convey it. Ask the questions you expect the receiver to ask you, so that you can answer them on behalf of the sender. Remove as many of your own assumptions about the message as you can.
- ***Leave your opinions at home***. When you are acting as an intermediary, your job is to deliver or translate the message as accurately as possible adding nothing to the message and taking nothing away. Express your opinions or reservations with the message when you receive it, not when you deliver it.

    Your deep understanding of the business mind-set and the organization's stance does not automatically give you leave to speak on behalf of the sponsor, customer, or organization. When you state opinions or valuations, they must be the voice of the customer, and not your interpretation of what you think the customer would say.
- ***Make sure the message is understood***. When translating, you have to make sure that the message has not only been received, but also understood. Make sure that there are no translation errors to exacerbate what might be an already delicate communication situation.
- ***Do not take a message you will have trouble delivering***. When you receive a message that does not seem right or that is too confrontational, or off target, it is better to deal with the sender than the receiver. Clear it up first and then deliver the result. Assuming that the content of the message is not your problem is making a dangerous assumption. Remember the classic fate of the messenger who bears bad news.
- ***Understand the expectations***. Each of the parties involved in making a change in the organization has expectations of the change. Each of the parties has expectations of the other parties with whom they are working. As an intermediary, you need to understand these expectations. When the expectations do not match reality as you know it, you need to do something to alter the expectations. A consistent flow of honest information about what is really happening helps keep expectations real.

Also remember that every party has expectations of you as business analyst, and those expectations might not be the same. As long as you can live up to those expectations, continue to march. When you cannot, you need to do some adjusting.

■ ***Deliver the message, don't just document it***. There is a temptation when receiving information in a face-to-face meeting designated for another party to relay that information to the other party via e-mail or other hard copy. This way you have a paper trail that you can refer to later. Because the documented message is permanent, you will naturally spend a lot of time making sure it is correct, politically and otherwise. Once a hard copy of a message is received, the recipient will also tend to respond in hard copy and start copying anyone who might be vaguely interested. The recipients will do the same. Then the e-mail floods engulf everyone's daily activities.

It is better to deliver the message in person (assuming geographical availability) or by voice so that you can receive immediate feedback and discuss the information to prevent misunderstandings and miscommunications. You can always document the transaction for the records later. A business analyst's time should be spent communicating not writing documentation.

■ ***Work yourself out of a job***. What you want to do is to get out of the role of just being a translator or intermediary. Educate both IT and the business to minimize the need for translation. Encourage the people on the solution teams and the rest of IT to meet face-to-face with the business and vice versa. Attend as a translator where necessary and as a referee when you must. Eventually, you will be able to spend your time in much more meaningful, interesting, and valuable roles.

Filter

The art of progress is to preserve order amid change and to preserve change amid order.

— Alfred North Whitehead (1861–1947)

The goal of the filter role is to receive requests for IT work from the business community and then analyze, consolidate, eliminate, and direct the requests to the appropriate specialists with a recommended solution. Done poorly, the filter may filter too much or filter nothing at all. Done well, the business analyst acts like a business internist.

Sometimes we, on the solution side of the business, get the impression that there is a large faucet dripping problems into the organization; an

unending, unedited flow of problems, change requests, opportunities, defect reports, and the like, emanating from the business. Sometimes the drip is faster than at other times. There are no decisions made about the validity or applicability of the problems; they just flow our way, unfettered and unfiltered.

The business analyst provides a filter for all the problems that come into IT's queue. The problems are received in the guise of change requests, trouble reports, policy changes, new regulations, edicts from upper-level management, and the electronic suggestion box. Some users have many problems; others few.

Many times the same problem is reported by a number of users in different ways. The requests must be consolidated, analyzed, and filtered to prevent duplicate or similar problem statements from generating redundant work. In the role of the filter, the business analyst contributes to the prioritization of the problems by noting the number of times a problem is reported and the scope of the problem in the user community, and by objectively evaluating the impact on the business. The stakeholders many times will not understand the true impact to the business, only the impact to themselves or their part of the business.

Giant Filter

As shown in Figure 5.1, one of the business analyst's jobs is to receive requests for change from anyone in the company, including management and IT. In this role, the business analyst performs the following activities:

- Evaluate change requests and defect reports for validity and completeness, assisting the submitter, when necessary, to complete the request.
- Direct requests to the pertinent IT area (such as networking, database administration, or software development).
- Prepare business case or other decision document for board or management review.
- Consolidate multiple requests into a single problem statement.
- Ensure that what is delivered to IT for a solution does in fact state the real problem and not simply a restatement of a set of symptoms.
- Eliminate specious or spurious requests.
- Identify problems in the business community that need addressing.
- Pass on to IT only those problems that have a positive cost/benefit to the organization; those problems that management has decided to solve.

The business analyst as filter solves a common change management problem for many organizations. For organizations where all IT requests went to individual areas of IT, the users had to know which IT department to submit the request to, or they submitted it to the help desk where it often
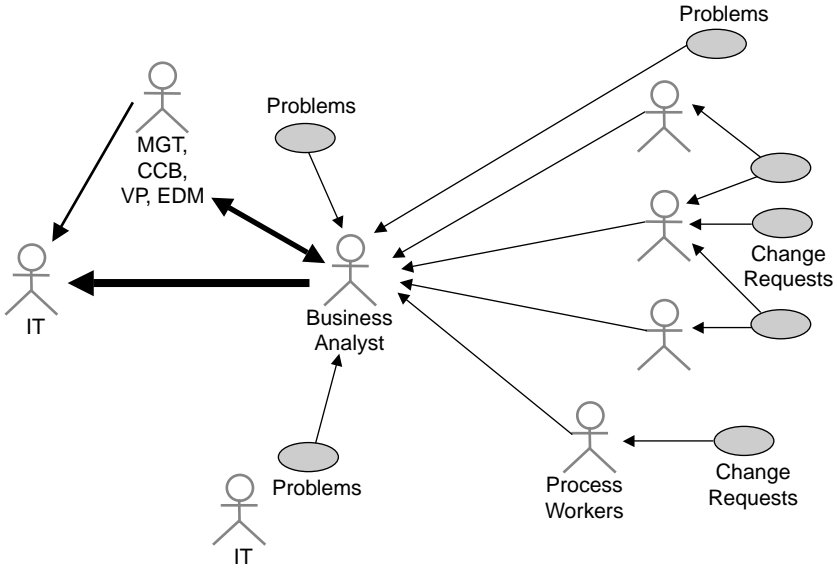
FIGURE 5.1 The Business Analyst as a Filter

got lost in the shuffle. After a while users got to know certain programmers or analysts, who had performed successfully for them, and forwarded all subsequent requests directly to that one person. When the change was not big, the programmer simply made it. There was no evaluation of the worth of the change to the organization. There was no evaluation of the impact to other systems or business processes. IT lost the capacity to do the larger projects because of the steady stream of smaller, user requests that occupied the developers' time.

Filtering Frivolity

I do not mean to impugn the business community by suggesting that there are process workers and users out there flooding the suggestion box and help desk lines with frivolous requests. The business community typically makes legitimate requests. Not all legitimate requests should be acted upon. The business analysts may determine that the legitimate requests are not actionable for the following reasons, among others:

- The request is for something that already exists.
- The requestor has a lack of understanding of a system or technology.
- The request is technologically based request rather than one based on business goals.

- It is a duplicate request of something that has already been done or is in progress.
- The request shows a misguided focus on departmental rather than organizational goals.
- It is a request from management that is not realistic when compared with actual operations.
- It is a request that cannot be aligned with organizational missions, strategies, or goals.

Repair Service Behavior

A squeaky wheel syndrome surfaces when the business analysts focus on trivialities, and real problems are hidden in the noise. The business analyst then focuses on reducing the list of issues received from the business community rather than solving real problems for the organization. Management may contribute by judging the business analyst group based on the size of its backlog rather than number of successful solutions completed.

In this environment, the business analysts fail to investigate connections and dependencies between complaints, trouble reports, and requests for change. The business analyst unit begins "setting priorities by the criterion of obviousness."[3] The business analysts address the problems they know how to solve first. The concept is to "get rid of the easy noise so we have time to focus on the important ones." Unfortunately the easy noise is never silenced.

This is what Dorner refers to as repair service behavior. "When we act on the basis of a more or less randomly generated list of complaints, we necessarily remain captives of the present moment . . . We should therefore take the future into account when dealing with dynamic systems."[4] This is the to-do list behavior that Stephen Covey[5] warns us about. In an effort to simply reduce the size of the to-do list we complete items that are easy to complete without regard to their importance or urgency.

⭐ Just for Fun

In the extreme, the filtering unit becomes so overwhelmed with demands that they only record complaints and issues with no analysis or review. A friend of mine who worked for a very large church organization related that the help desk, which was populated by business analysts assigned to the filter role, had such a backlog of unanswered phone requests for changes and defect fixes that the help desk was known as dial-a-prayer.

It is easy for the business analyst to get lost in a never-ending series of business problems and requests. A business analyst acting as a filter without applying the role of investigator and analyst is just sorting and rearranging stakeholder requests.

Mediator

"How do I handle conflict between my project team and the users and sometimes [name of the customer]?"

Some organizations use a project management office (PMO) to help resolve issues between and among projects, primarily because the conflict generally revolves around resource allocation disagreements, conflicting technical approaches, or use of facilities. Some PMOs get further into the details of each project and resolve issues of conflicting requirements among business constituencies.

In organizations that do not have a PMO or where the PMO does not get into interproject and interdepartment disputes, the business analyst becomes the default mediator to resolve issues such as:

- Conflicting requirements.
- Indecision amidst business groups.
- Differing approaches between IT and business to solving a specific problem.
- Authority disputes between departments.

The role of mediator is similar to the role of referee. The referee is limited to reminding the participants of the rules and enforcing the rules when applicable. The mediator does the following:

- Establishes ground rules.
- Slows down the conversation.
- Clarifies points providing translation where necessary.
- Asks questions to promote understanding.
- Suggests alternative solutions.
- Facilitates both parties to come to a mutually beneficial solution.

The mediator does not make any decisions or judgments on behalf of either party.

Why Mediate?

You are involved in mediation when there is a perception of conflict between parties and at least one party is afraid to talk to the other party due to:

- Imagined negative consequences of discussion (i.e., the developers fear users will add more functions to the requirements; the users fear the solution team will reduce the number of delivered functions to meet the project deadline).
- Historical tapes playing about previous meetings of this nature, perhaps not even in the same organization ("You can never talk to those people; they never listen").
- Political concerns about having meetings without certain players.
- One or the other party is too close to the situation.

Mediation also takes place when two parties will not come to an agreement on their own because it might:

- Be admitting defeat.
- Be embarrassing to admit their position is not absolute.
- Be a power thing.
- Require more work.
- Be a sign of weakness and not bode well for future negotiations.

Role as a Mediator

The role of the business analyst during mediation is that of a facilitator making it easy for the two parties to work out a conflict or perceived conflict. The most important attribute of the mediator role is objectivity: You cannot be perceived as on anyone's side.

When should an analyst take on the role of mediator? Table 5.1 addresses this question briefly.

TABLE 5.1 When to Take the Role of Mediator

When to take on the role of mediator as a business analyst	When to not take on the role of mediator as a business analyst
The business analyst is already in the center of things anyway.	When the decision is made, the business analyst may be perceived as taking sides.
The business analyst knows all the parties and usually all the issues.	The business analyst is new to the area and does not know the parties or the issues.
The business analyst is known to be objective and neither a representative of the business nor IT.	The business analyst clearly represents or is perceived to represent IT or the business.
The business analyst is a problem solver and what is being mediated is a problem.	There are personal relationships at stake between the business analyst and one of the participants.

To be effective when you are called upon to mediate, first play the role of investigator. You identify the point of contention, then the position of each of the parties. When you talk to each party, they will usually point at the other party as the cause of the contention. You need to find out what each party wants. Then you bring the parties together face-to-face or voice-to-voice. Avoid a mediation-by-message situation, where you become the messenger.

Your goal in mediation is to orchestrate a solution that gives both parties what they want, if possible, and increases the chances that both parties will work together for the duration of the solution life cycle.

Diplomat

Because you are in the center, you are constantly negotiating regardless of your desire to do so. It is a role that comes with being a business analyst. Sometimes your success as a business analyst is dependent on your success at diplomatically bringing diverse groups in the organization to a mutual understanding and resolving interdepartmental conflicts that arise whenever anything in the organization changes.

> *Tact is the art of making a point without making an enemy.*
> —Howard W. Newton

The business analyst is in the center of the story, which means that you are privy to all sorts of information, perhaps more than anyone else in the process. The stakeholders may look to the business analyst to negotiate for them with other parties.

> *In the life cycle of every conflict, there is a point when it's large enough to be recognized, but small enough to be resolved.*
> —Anonymous

Arbitration and Conflict Resolution

As a business analyst, you are in a continual state of negotiation. You are negotiating for time. You are negotiating to get people to meetings. You are negotiating with the business about scope; you are negotiating with management about scope; you are negotiating with the project manager and team about scope. You are negotiating among business units to resolve conflicting expectations. Sometimes you will negotiate on behalf of the business community, sometimes on behalf of the solution team, sometimes for management. You are, after all, in the middle (see Figure 5.2).
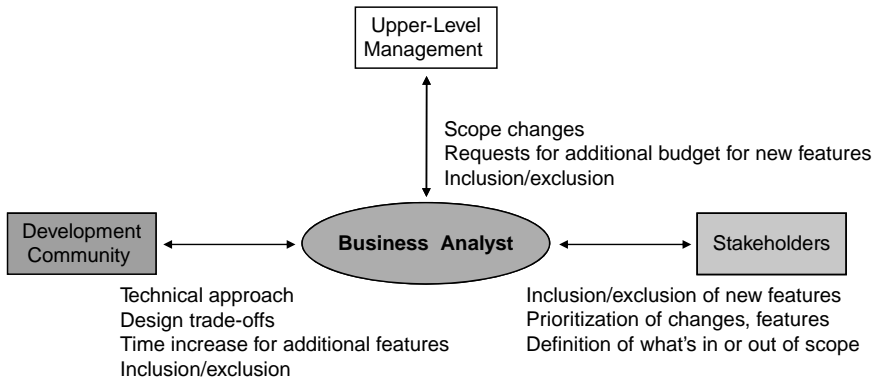
FIGURE 5.2 The Business Analyst as Negotiator

The business analyst strives to keep all parties developing the solution focused on that solution. You will use many forms of negotiation: hallway discussions, formal meetings for the purpose of ironing out issues, presentations to a decision maker to render a judgment, impromptu interruptions of a gathering convened for another purpose, a conference call, a string of e-mails, and so forth.

Collaboration Is the Key

A successful collaboration "has a clear goal and well specified expectations. All the team members understand why they are working together." Each team member has a "well-defined role and trusts others to fulfill their roles." The collaboration also "has buy-in from all appropriate levels in all the participating organizations."[6]

The project manager is responsible for creating a collaborative effort internally within the solution team. The project manager should also take the lead in any collaborative effort among other projects in progress that may share resources or be working in the same area. A PMO is also a good instigator of collaboration among projects when an organization has a PMO in place.

The business analyst creates collaboration among:

- The business community and the development community.
- Various business communities who may have a partial stake in the results.
- Entities outside the organization that may be impacted by the change.
- Other members of the technical community who may assist in solving the problem.

Despite all the inherent and obvious benefits of collaboration, getting successful collaboration is not easy. Start with the concept that you are going to create a collaborative environment, and diplomatically resist and overcome the objections to collaboration:

- Distrust.
- Desire to withhold or suppress information.
- The win-lose mentality common in business, especially in the United States.
- Lack of geographic collocation.
- Departmental infighting.
- Political intrigue.

Especially in cases where the business community or the solution team is spread out geographically, the business analyst typically becomes the central point of collaboration. Introduce collaborative tools into common use by the business community as well as the solution team. Collaboration tools offer functions and features that enable collaboration locally and around the globe and include:

- Shared calendars.
- Meeting scheduling.
- Online meeting execution.
- Project or team workspaces.
- Discussion areas.
- Repositories of all problem and solution-related materials and so forth.

"What can I do to increase collaboration among all the parties in the solution development effort?"

Start by increasing the number of informal meetings. I know—no one likes meetings. We're not talking formal, agenda-driven, take-copious-notes-just-in-case, why-are-we-here meetings. During a conversation with a developer about a particular function, get the user on the phone and conference her in to answer some questions. Have the testers join you when you are eliciting information and defining the solution to make sure the requirements you define are testable. Conduct impromptu discussions in the hallway or the coffee room or at lunchtime or even at happy hour. The concept is not necessarily obtaining information or decisions, but increasing the interaction among the varied team members, which will increase collaboration in general.

While it is beneficial to include a wide range of process workers in your elicitation to get exposure to a wide range of ideas and concerns, try to

select specific individuals as the primary representatives. Collaboration is easier with a smaller group.

As much as possible, expose the technical personnel to the business drivers behind the changes and other aspects of the business and expose the businesspeople to the intricacies of both the technical solution and the process of implementing that solution.

Create and maintain a public knowledge base about the problem and solution. The more information that is shared among the participants in the solution effort, the greater the collaboration.

Collaboration by Documentation

Despite your most ardent efforts to increase the flow of face-to-face and voice-to-voice communication, some on the project team and management may opt for documents as their primary source of communication and collaboration. You can increase the collaborative value of the documents you are forced to prepare by first asking the recipient what information they need in the document and why they need that particular information (i.e., what are you going to do with this information?). Disregard formal templates and standard formats and ask them what they really need to know and when they need to know it. What decisions are they going to make based on the information you provide? Then prepare the document according to their specifications. In this way, you are increasing the probability of getting feedback from the document that will, in turn, increase the collaboration, even when the recipient does not actively collaborate.

Politician

> *Politics is about working and negotiating with others in your organization to get things done.*
>
> —Chad Dickerson, CTO, InfoWorld

Politics is neither good nor bad. The business analyst needs to be aware of the political situation at all times. There is nothing magic to understanding corporate politics. You just have to recognize the motivation. Every position each person takes in relation to a controversy is motivated by something. Once you understand the motivation, the politics are clear.

Focus on the problem and the solution. Remain objective and outside the politics that typically surround any change that takes place in the organization. You should be able to rightfully say, "I'm not on anyone's side. I'm here to solve this business problem." Politics is mostly about the judicious

application of information, which is done by withholding it sometimes and releasing it other times.

Dealing effectively with political pressure means remaining objective, focusing on the problem to be solved, and keeping the information flowing. You certainly have an opinion and you believe in the solution you have devised to solve the problem. However, your goal is to get the problem solved. The final solution may not be the one you recommended. The choice may have been made strictly for expediency rather than efficiency or elegance. Ultimately, the only thing that is important is that the problem is solved. Separating yourself from the politics may not be easy, especially when one of the players is your boss. You cannot eliminate or even reduce politics. You can lessen the impact of the politics on yourself by focusing on the problem, vision, and solution.

Investigator

Problems are like mysteries. To solve problems or mysteries you need information and to get the necessary information you need to investigate. Like a detective, the business analyst conducts investigations to determine the real problem and the best solution. Like a detective, the business analyst applies her powers of deduction to determine the solution. And, like a detective, the degree of success is usually predicated on the questions that are asked and the information gathered.

> *Doubt, indulged and cherished, is in danger of becoming denial; but if honest, and bent on thorough investigation, it may soon lead to full establishment of the truth.*
>
> —Ambrose Bierce

Central to the business analyst's professional existence is a business problem. The business analyst's job is to solve a problem. Communication, elicitation, analysis, testing, and documentation are all about solving that problem. Along the way the business analyst solves a great many other problems with negotiation, mediation, action, reaction, suggestions, and so forth. The business analyst's focus from beginning to end is always on the problem and its solution.

The business analyst follows the same process as the classic investigators we are all familiar with: Columbo, Lenny Briscoe from *Law & Order*, Hercule Poirot, and, of course, Sherlock Holmes. Establish the crime (business problem), ask questions and gather information, analyze that information, use technology to gather more information, diagram the information, and then deduce the perpetrator (deduce the solution to the business problem).

According to U.S. Department of Labor Bureau of Labor Statistics, "Detectives and investigators assist individuals, businesses, and attorneys by finding and analyzing information. They connect small clues to solve mysteries or to uncover facts about legal, financial, or personal matters." The business analyst spends quite a bit of time investigating things, asking questions, looking for clues, and examining evidence.

The business analyst is a solution-oriented role that identifies the real problem and determines the solution. The business analyst collects what may appear to be unrelated facts, perceptions, opinions, his own and others' observations, inferences, and his own experience and knowledge to determine the real problem and the best solution. The business analyst looks for order beneath the confusion, a logical flow of work within apparent disparate activities, meaning in the day-to-day confusion. It takes a lot of persistence, attention to detail, and the ability to filter out nonessential data. The difference between the business analyst and the detective is in the end result. The detective solves the crime and turns the criminal over to the authorities. The business analyst determines the solution and documents it in a solution document for those who need the problem solved and those who are going to implement the solution.

In successful deduction the business analyst applies reasoning based on general or universal principles to arrive at a conclusion that cannot be false unless the premises are false. The deduced conclusion is of equal importance as the information on which the conclusion is based. The premises or principles are the business rules, policies and procedures, operational guidelines and descriptions, confirmed statements of fact from those that use the process and so forth.

> *Project teams must continually ask and answer two questions: "Which business problem are we trying to solve?" and "How are we going to measure our solution's impact on this problem?" In my experience, these two questions are not asked often enough or with sufficient objectivity.*
> —Troy Kinsey, UC Berkeley, *Computerworld*, 6/13/05

Analyst

> *Analysis is frustrating, full of complex interpersonal relationships, indefinite, and difficult.*
> —Tom DeMarco

According to the Oxford English Dictionary, analysis is defined as "the process of breaking a concept down into more simple parts, so that its logical structure is displayed."

Analysis sets the business analyst apart from the requirements recorder. And the role of analyst is where many business analysts fail in their problem-solving goals. Unfortunately many BAs perceive their role simply as the documenter of requirements statements from purported subject matter experts, and performing little or no analysis on the content.

The key to analysis might be skepticism. The business analyst does not accept information at face value. As discussed, during investigation, the business analyst seeks confirmation of the information obtained. As an analyst, you:

- Challenge assumptions.
- Question business rules.
- Look for alternate solutions.
- Validate or confirm information.
- Evaluate potential solutions.
- Examine existing business processes.
- Ask the hard questions and question the answers.

What you are looking for are the logical connections, the patterns, the flow of information, and then you are looking for the anomalies and disconnects, the inefficiencies and the redundancies. What is wrong? What is right? And above all, why?

The business analyst analyzes the information obtained during the investigation to define an acceptable solution to a business problem and describe the characteristics of that solution such that the solution team has a clear understanding of how to design and implement it.

The business analyst goes through two general phases of analysis when solving the business problem. The first is analyzing the problem, which consists of combing through gathered data to determine the needs of the user. The second phase is the analysis of the possible solutions to determine which solution is best based on impact analysis, gap analysis, and the feedback from the product stakeholders. And, of course, there are budgetary, regulatory, and political constraints on the choice of a best solution that the business analyst has to keep in focus as well.

When solving the problem, the business analyst draws on his analytical experience and skills to make deductions about:

- Functional goals and their attributes.
- The appearance of an anomaly and why the anomaly appeared.
- The urgency of solving a problem and the benefit of doing so.
- What constitutes an opportunity worth pursuing and how to prove that it is worth pursuing.

- Which analogues best fit the solution and how to apply them.
- The best solution to the defined business problem.

Change Agent

The problem is not solved until the solution is installed in the business production environment and is being used by the process workers. Even when the project is successful—on time, within budget, with all requirements satisfied—the product may not be accepted in the workplace. The business analyst has to make sure there is a smooth transition from the current state of operations to the problem-free state so that the product is used to solve the problem.

> *Advocates of progress often have too low an opinion of what already exists.*
>
> —Bertolt Brecht

A change agent is one who advocates for an innovation in the organization. The business analyst assumes the role of change agent to coordinate the implementation of the change in the business community. The business analyst is explicitly or implicitly responsible for the successful adoption of changed processes, products, and technologies in the organization. Beyond simply transitioning the organization to the new problem-free state, the business analyst drives the change itself by identifying what changes need to be made to improve the overall business processes and add value to the organization.

As agents of change, we need to:

- Define the changes to be made (the problem to be solved).
- Assess the impacts of those changes on the organization as a whole.
- Determine the ability of the organization to absorb and apply new changes and create a schedule of change that allows for successful adoption.
- Prepare the business community (process workers and managers) for the transition.
- Confirm that the change has been made successfully in the production environment (the implemented change is in fact solving the business problem).

It is not enough to just be an advocate for the new product. As a business analyst, you have to understand the business community and its ability to absorb the change. Even a change considered to be minor, such as adding

a check box on a data entry screen, may bring about trepidation to some process workers. Conversely a major change, such as replacing the in-house accounting system with a purchased product, might be taken in stride by everyone. The difference is in the way the change is presented and how we orchestrate the transition from old to new.

Quality Control Specialist

The business community expects that the business analyst is going to produce a solution for their problem. You do not want to disappoint them by delivering a product that works fine according to the specifications but does not really solve their problem. Testing the solution is the way of making sure that the expected level of functionality and quality are present in the solution when it is placed into production.

> *Measures of quality have to go beyond whether the system is meeting specifications to ask if the system is meeting user's expectations.*
> —Peter Coffee, eWeek, 5/6/06

Most business analysts write acceptance test cases, manage the users in executing the test cases, execute the test cases themselves, or assist the Quality Assurance Department in doing any or all of these tasks. Some business analysts are involved with system testing and a few are involved in integration testing. The BABOK includes testing as a common role for business analysts, although it is a role not within the scope of the BABOK itself.[7]

Ensuring the product solves the business problem before it is put into production is a matter of self-preservation or perhaps personal pride. As a business analyst, you work with the business and help the business define the problem. You gather information, analyze the information, and determine a solution. You present that solution to the business and they pronounce it good: "You do what you have specified in these requirements and our problem is solved!" Now you turn the solution over to the solution team.

After a while the solution is ready. The solution team runs a series of tests on the software and hardware solution: software unit tests, integration tests to put the software and hardware together, and system test to check that the system works with all the quality attributes specified.

Quality assurance steps in to run a set of predetermined acceptance tests, and the solution is pronounced ready. Since you are the face of that solution to the customer and users, you want to make sure that the problem you promised to solve is actually solved.

To provide your own assurance that the problem is solved with the requisite quality, you are going to fashion some tests that prove to you and

the business that the problem is, in fact, solved. You know that there are a great number of things that can happen during the software development process, even when the customer is involved. Everyone may go off on a tangent. It may be a valuable tangent, but it may not solve the original problem. As a business analyst, you have committed to the business to solve the original problem unless and until the business decides the problem no longer needs solution.

Whether you or QA run the tests to prove the problem has been solved, you want to make sure that the results of the test prove the solution team has done what it set out to do, and what the business community expected.

How do you know that the tests you devise will prove this to the customer? You asked. You asked the customer or the problem owner what it would take for them to agree that the problem is solved. You asked: what do they need to see to believe that we solved their problem?

Facilitator

Everything the business analyst does is about making things easier for all parties involved in the process of solving a business problem. This is the facilitator role. Much of the facilitator role occurs automatically when other roles are played. While other roles a business analyst plays have impact on the processes and functions of the organization, this role is the one that has the greatest impact on the people in the organization.

> *More people see how you deal with people than will ever really know what you do.*
>
> —Chad Dickerson, former CTO InfoWorld

According to Wordnet.com, the word facilitator means "someone who makes progress easier." Dictionary.com defines the word as "a person responsible for leading or coordinating the work of a group." Also, according to Merriam-Webster, the facilitator is "one that helps bring about an outcome (as learning, productivity, or communication) by providing indirect or unobtrusive assistance, guidance, or supervision." The business analyst is a facilitator in all these meanings, leading a coalition of business, IT, and management to solve the business problem, making it easier for everyone to progress toward a solution.

The Business Analyst's Handbook from a public utility company (compiled by a highly effective IT organization) says, "Your job as a business analyst is a role of facilitation." The *Handbook* goes on to describe how the facilitation is achieved:

- Providing a project vision and high-level scope so that automation project can be approved and initiated,
- Gathering and documenting business and user requirements,
- Translating those business requirements into a set of business rules (business use cases),
- Development of acceptance test plans and execution of those plans for client acceptance of automation products,
- Application training and mentoring,
- Assisting the customer area with process and procedure development as it relates to automation,[8]
- Performing IT product needs assessments from a business perspective,
- Developing business process mapping and workflow documentation,
- Consultation on potential business uses of automation,
- Client contact for production support,
- Second-level end-user help in using automation,
- Business implementation planning,
- Managing maintenance tasks that do not require the direct involvement of the project manager,
- Assisting customers with defining enhancements to existing systems,
- Providing ad hoc reporting for customers.

Facilitating Upper-Level Management Decision Making

The business analyst assists upper-level management in making strategic and tactical decisions by providing the information to decide on strategy issues such as mergers and acquisitions, divestitures, new product lines, reorganizations, and so forth. Business analysts also spend a lot of time providing the information that justifies a project, such as cost/benefit analysis (C/BA), return on investment (ROI) analysis, and feasibility studies. The business analyst is increasingly instrumental in one of the more important decisions upper-level management makes: determining which vendors should be chosen to perform specialized work for the organization, or provide the goods and services the organization needs to survive. This is done through a request for proposal process and in many organizations the process is managed and executed by business analysts.

Good Information Makes Good Decisions

The primary cause of bad decisions is insufficient or incorrect information. Decision makers need good and complete information to make their

decisions. They need to know what the consequences of their decisions will be. This is where you, the business analyst, come in. You provide the information to help management make their decisions. You do the research, organize the information, filter out the unnecessary noise, identify the potential solutions, and so forth. And you present the whole package in a structured method:

- Here is the problem (or decision to be made).
- Here is the risk if we do not solve the problem or make the decision.
- Here are the possible solutions.
- Here is the risk for each solution.
- Here is the time necessary to implement the solution.
- Here is the timeframe within which you must make a decision.
- Can I have a decision?

Providing information in this manner allows decision makers to make appropriate decisions.

Where the Business Analyst Can Help

The following are various different strategic and tactical efforts that the organization may decide to undertake. In every one of them, the business analyst may provide information, metrics, comparisons, ROI ratios, cost benefit analyses, studies, and so forth to help upper-level management make decisions that affect the entire organization, for example:

- Major reorganizations.
- Improvements to the organization's competitive position.
- Implementation of total quality management or some other quality guideline.
- Improvements to customer service attitude and behavior throughout the organization.
- Acquisitions, mergers, or divestitures.
- New product or product line to the product mix.
- Compliance with new government regulations or industry guidelines.
- Changes to policies regarding employee benefits and workplace.
- Reduction, expansion, or relocation of the workforce.
- Cost reduction or cost containment programs.

It isn't necessary to have an opinion or a great deal of knowledge about any of these decisions. The business analyst's job is to facilitate the decision making by acquiring the information on which a reasoned decision can be made.

Buy, Don't Build

Not every solution is built internally by IT. Many organizations are opting for the buy route of obtaining the solution from outside the organization. There are two alternative buy routes to take:

1. Buy the solution off-the-shelf (OTS) or commercial off-the-shelf (COTS).
2. Hire a vendor to develop the solution and deliver a turnkey package.

Both alternatives have advantages over building the software in-house. Many organizations are reevaluating their investment in internal software development and deciding that building software application systems is not their core business so they are going outside. Other organizations are parceling out parts of their IT operation to offshore developers as a way to save money and reduce management overhead.

The business analyst is instrumental in the decision to build or buy. When the decision is made to purchase, there is usually a request for proposal (RFP) for vendor-developed software or a request for quote for OTS alternatives. In the process of obtaining an outside vendor or OTS solution, the business analyst usually has these responsibilities:

- Identify the potential sources of the solution (make sure there are enough potential vendors for competition).
- Prepare the RFP or RFQ defining the requirements that the vendor must comply with to be selected.
- Work with the purchasing department or vendor relations to help administer the bidding process (manage the bidder's conference, respond to bidder's questions, etc.).
- Evaluate the technical portion of proposals and bids against the requirements providing a ranking to management.
- Orchestrate any demonstrations of vendor's products or test the software package and provide evaluations of the products.
- Identify the shortcomings and gaps in all vendors' proposals and bids and be prepared to offer alternative to filling the gaps should the vendor be selected.
- Close out the process after management makes the decision by notifying all bidders.

Facilitating the Business Definition of the Problem

Often users or customers have only vague ideas of the problem or need. The business analyst can help the process worker make that goal specific and overcome fears about committing to a course of action.

You help the business define the problem and the solution that will work for them by:

- Tactfully challenging their assumptions about the way things are.
- Tactfully challenging their assumptions about what can be done to solve the problem.
- Educating them as to the technological possibilities available to solve the problem or improve the situation.
- Encouraging them to take a more holistic view of the overall problem domain to evaluate impacts outside the target zone.
- Keeping them involved in the investigation and analysis so they can see the solution develop and provide feedback along the way.

Facilitating the IT Definition of the Solution

As a business analyst, you can make it easier for the solution team to determine how to design the system, code the programs or objects, connect the components on the network, arrange the data in the databases, and so forth, by explaining clearly what the business problem is, why the business needs the solution, what the best solution for the business looks like based on the current information, and what the organization will gain from a good solution. This is all part of the problem definition process we discuss in Chapter 8.

🔍 Example

In my early days as a programmer and for many years thereafter, I wrote programs based on a program specification or some other document that told me what the output of my program was to be and provided the description of the input and algorithms I was to use to produce that output. I produced reports for people I never met with no idea of how the information on the report would be used. I put the results of my work into a box and then received any feedback in the form of marked-up paper. This was not all bad since the report was the primary user interface at the time (all input data came from punched cards) so communication in this fashion worked all right. Occasionally I was asked to speak to the recipient of the report, and in doing so learned a bit about why the report was necessary and for what it was used. I found that when I engaged with the user the programming job became more interesting. It was as much about solving a problem and helping someone out as it was about moving bits in and out of registers. In some cases, I even

made suggestions that enhanced the information for the user. I do not know that the report format was any better or that my code was more precise. I just felt better about doing the coding when I knew the results were going to be beneficial to someone.

Process Improver

The business analyst adds value to the organization by solving problems. The business analyst also adds value by identifying areas of improvement in the business, processes which are no longer valid or applicable, redundant or extraneous processes, suboptimized activities within processes, and so forth.

> *When someone says "There are no rules here," what he means is "The rules are so set that nobody notices them anymore."*
> —Irwin Shaw, Bread upon the Water

Certainly the first job of the business analyst is to respond to the cries of pain from the business when problems are encountered. The business analyst also looks for additional problems in the business processes being investigated and solved to prevent future cries of pain. These problems are areas of the company where improvements can be made to save money, increase productivity, increase sales, and so forth. Julian Sammy, Chief Architect of the IIBA, calls this role enterprise business analysts.[9]

The business analyst is in the best position in the organization to see what is actually going on in the processes and operations of the organization. The business analyst can see:

- Failures in IT projects and the general reason for the failures.
- Symptoms instead of problems, and problems instead of symptoms.
- The larger picture.
- The actual results of product development.
- A holistic view of the business community and the organization as a whole.

By measuring before and after, checking out the request and the result, the business analyst will be able to tell whether the request was justified in the end, and whether the result was what was requested.

Increase the Value of Organizational Business Processes

> *Any creative person has to try and force their brain to reconsider things that are accepted so widely they seem like laws of the universe.*
> —Alan Kay

"How can we get ahead of our customers so that we can anticipate what they want?"

Whatever else you do as a business analyst, your prime responsibility is to increase the value of your organization by improving the organization's business processes: making them more efficient, more effective, less labor intensive, more flexible and responsive. Each improvement, each problem solved, no matter how small, increases the value of the organization. And that is the business analyst's job.

> *"Don't fix it if it ain't broke" presupposed that you can't improve something that works reasonably well already. If the world's inventors had believed this, we'd still be driving Model A Fords and using outhouses.*
>
> —H. W. Kenton

While the business analyst is solving a stated business problem, he or she is evaluating and defining the entire problem domain. In doing so, they see other problems in the business processes that surround the defined problem. This increases overall value to the organization.

So what is value to the business? It is something that delivers profit to the organization by increasing revenue, decreasing cost, or improving service.

Value is achieved by improving the quality of the information that the process workers use, and by making more information available to each worker. The value of IT's infrastructure is a result of the value of the information it makes accessible.

The message here is simple. As much as the sponsor or the project manager might like you to have laser-like focus on generating the requirements, the business analyst's first allegiance is to the organization. So, like the aforementioned investigators who, while investigating a break-in uncover a broad conspiracy, the business analyst must follow up the clues to uncover and resolve larger issues that are reducing the value of the organization. Politics may come into play since the project manager has a deadline to deliver the goods and they may consider anything else out of scope no matter how critical it might be to the health of the organization, and the sponsor, who is only concerned about his or her problems, may support the project manager. However, the business analyst's obligation is to keep their eyes open and at least report their findings to someone who needs to know, even if they are not able to act on those findings.

Build It and They Will Come

So you defined the solution and the product was built. Then the question is, "How will the customer and/or the business know what value has been

added?" Value is felt as well as measured. You might be able to show with measurements that the response time has improved by 20 percent and the users might still not feel that they have obtained value for the cost of that improvement.

A friend of mine from Tennessee told me about a bank he had worked for a number of years ago. The IT department decided to add value to the bank's business processes by building a data warehouse containing all the bank's customer data, financial records, historical records which the bank had to keep for legal reasons, and so forth. They envisioned tellers and branch managers and loan officers being able to make decisions quickly and accurately: who should get mortgage loans and commercial loans, where to invest the bank's money, when to buy and when to sell, which bank customers should be courted, and so forth. After two years and much money spent, the warehouse was built. There was much discussion with the bank staff in the beginning to determine what kind of decisions were made and what kind of information was needed to make those decisions. When the data warehouse was ready for use, thoroughly tested, and enhanced with the latest business intelligence software to create graphs and charts and do pivots and drilling, no one used it. One of my friend's tasks had been to write the software to measure the usage so that IT could show management the benefit of the data warehouse. After a few months, he was ordered to fix his software because the numbers were clearly wrong. Few were using the warehouse and most who were using it generated reports that they could already get with the existing systems. Eventually the project was abandoned because it was costing more to maintain the system than the benefit derived from it.

What had happened? No one made the business community aware of the value of the product. Both my friend and I, having had experience with fully operational and well-used data warehouses, know the huge value a warehouse can bring to an organization. The employees of this bank clearly did not know.

The bank at the time did not have business analysts. No one investigated the issue with the process workers, no one established the value of the product in the business, no one sold that value to the business, and no one orchestrated the transition so the process workers understood the value.

It is not only about training or user's manuals. It is not solely about sales. It is about including the business throughout development to create excitement for the change. The value is there. The business just has to be shown. When the value is not there, the product should not be built. And the business analyst does add value to the organization by pointing out projects that should not be run and problems that should not be solved.

Reducing Complexity

Systems, especially computer systems, are complex. With each modification and upgrade and release of new software, the systems get more complex. Each new feature and function adds complexity by offering more options and alternatives. Reducing complexity adds value. There is less chance for human error. Less time is spent deciding among options. There is less hardware, software, and process to maintain.

There is always a limit on the amount of productivity for which an organization can pay. There are diminishing returns at some point, where increased productivity does not translate into sufficient increased revenue to cover the cost of the change. There are also points at which increased productivity does not mean increased quality. The business analyst understands the organizational strategy and helps determine how much added value the organization is willing to pay for.

Many times the workers become like children at Christmas when asked what they want under the tree. Workers can think of all sorts of snazzy features that would be nice to have. Some they have seen on the Web, some they have at home, some they have read about in airline magazines, and some are just fanciful notions.

The question to ask is whether a particular feature will bring value to the people using it and ultimately to the organization. How often will it be used? Who will use it? Will it be used instead of an existing feature to get a job done better or faster, or will it add more work to the users? Will it be so complicated and cumbersome that it will actually cost more to train and keep up, requiring additional people on the help desk?

This is where the agile concept of incremental delivery comes into play. By delivering the features or functions that have the highest business value first, the business users may discover that they don't really need some of the doo dads they envisioned in the initial throes of defining the new system. They may determine that the sizzle is not worth the extra money to pay for it. The result is a less complex system overall and one that, indeed, meets the organization's needs.

The business analyst's job is not to simply react and figure out how to give the business whatever it wants. The business analyst's job is to increase value for the organization.

As business analyst, you contribute to either increasing or reducing complexity. When you accept all new features and changes to existing systems without evaluation and alignment, you are increasing complexity, especially when changes are made without examining the overall business process.

Playing Multiple Roles

I can (I have established from empirical evidence) do anything. I cannot (I have established from bitter experience) do everything.
                                                                    —P. Abraham, 1991

As a business analyst you will be playing all the roles described in this chapter for any given project. So what can you expect in terms of which roles you'll play and how much of your time will be devoted to each?

Remember that these are roles and not jobs. Each role has associated activities and responsibilities as described in the upcoming chapters. The business analyst may be playing multiple roles simultaneously, such as enterprise communicator and change agent. Some roles are played almost continuously and are the trademark roles of the business analyst (as rugged cowboy was the trademark role for John Wayne). Problem solver is one of these.

The important aspect of defining roles is to examine your own abilities and talents and determine which roles are easy for you to play and which are difficult. Identify which roles your organization expects of you as a business analyst, then focus on specific activities you perform when playing a particular role.

Once you have done an inventory of what you are good at doing and what is expected of you by the organization, you can make a decision to:

- Focus on the roles that you do well so that you continue to do them well and improve on your performance.
- Focus on the roles that you are weak in to bring them up to a level that will be acceptable in the organization.
- Change organizations because their expectations and your talents are not the same.
- Change professions because your talents do not fit any of the roles identified in this section.

Regardless of the choice you make, you are analyzing the problem and providing a solution: the essence of the business analyst.

Now that we have established who the business analyst is and what the business analyst does, the question becomes where does the business analyst perform these tasks or with whom do they work? Since business analysis is not done in a vacuum, how does the business analyst get collaboration, with whom does she communicate, who are the parties in negotiation and mediation, and so forth? Part Two describes the relationships the business analyst has with those who are part of the overall business analyst solution process: the players.

Notes

1. International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge*, version 2.0 (March 31, 2009), 6.

2. International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge,* version 1.6 (2006), 8.

3. Dietrich Dorner, *The Logic of Failure: Recognizing and Avoiding Error in Complex Situations* (New York: Basic Books, 1997), 60.

4. Ibid., 60–61.

5. Stephen Covey, *The Seven Habits of Highly Effective People* (New York: Free Press (Simon & Schuster), 1989).

6. Mary Loomis, ''Managing Collaboration,'' *Software Development* 79–80 (January 1998).

7. International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge,* version 2.0 (March 31, 2009), 10.

8. Excerpted from internal corporate business analyst handbook for an East Coast utility company, published 2006.

9. Julian Sammy, ''Different Roles for Business Analysts,'' IIBA Executive Blog (January 6, 2009).

The Players

The process of solving business problems is not one that is done independently or alone. Even identifying the problem and the changes necessary to solve the problem requires the participation of many levels of the business including management, both mid- and upper-level, as well as the process workers. Implementing the solution requires participation of the solution team comprising the manager of the team, the systems analysts, the developers, and other technologists. And then the transition of the changes into the organization touches a large cross section of the organization. As discussed earlier, the business analyst is at the center of it all, the eye of the storm, so to speak, and links all the various personalities together to form a team focused on the various aspects of solving the business problem. The systems approach to solving business problems requires the collaborative effort of many people playing many roles, and the business analyst is the one who guides the collaboration. This is not an easy job, considering the vagaries of organizational structures and the people in them. To be successful in organizing the solution, the business analyst needs to understand the basic relationships involved in the change process.

In this part, we investigate relationships: relationships of the business analyst with each of the primary functionaries in the solution life cycle and how each of these functionaries plays a part in solving the problem. We also explore the relationship with the project manager and systems analyst, especially the separation of duties. The roles represented in this part are the people who populate your process.

"There is no single point of responsibility for documenting and maintaining all the communications between business and technical teams about the project and requirements."

The single point for communication in all cases is the business analyst, whether the communication is documented or not. In the following illustration, there are arrows connecting the seven major constituencies that the business analyst deals with in the process of defining and solving the business problem. Some of the constituents, such as project manager and problem owner, require constant communication and a strong relationship, others, such as the executive decision maker and IT management, may only require an occasional notification. Regardless, all are in the loop and have an effect on the final product. Each of the arrows connecting the business analyst to the constituencies is bidirectional. Can you define what is communicated along that arrow for each of the constituencies? For example, the process workers provide information and confirmations to the business analyst and the business analyst asks questions and provides the process workers drafts of problem domain and solution documents.
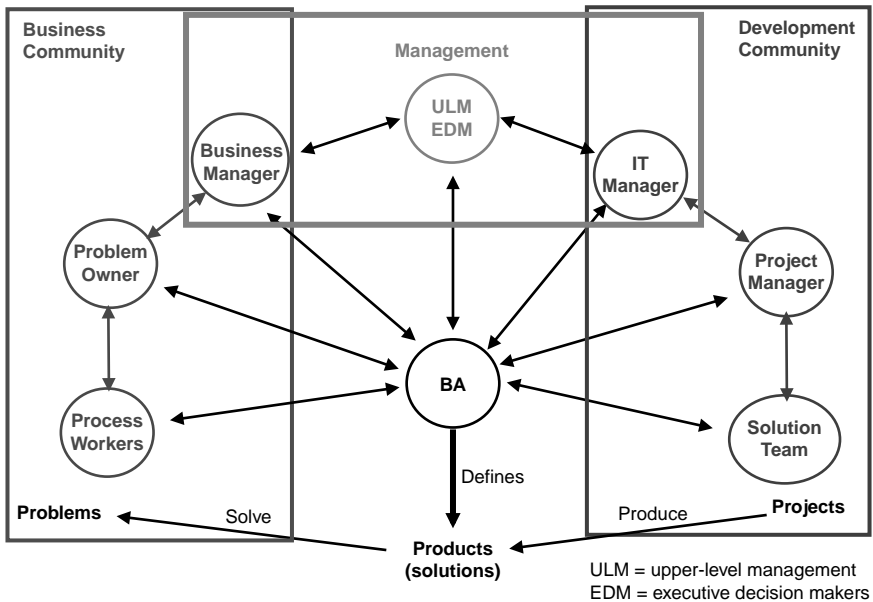
Business Analyst Relationships

The most important single ingredient in the formula of success is knowing how to get along with people.

—Theodore Roosevelt

You need to have good relationships with everyone in both the problem and solution domains. The amount and quality of the information on which you are basing the solution is affected by the relationship you have with those providing the information. Your relationship with the solution team may exert a great influence on their motivation to solve the problem, and provide guidelines that enable them to solve it faster and with higher quality.

CHAPTER **6**

The Business Analyst and the Solution Team

The meeting of two personalities is like the contact of two chemical substances: if there is any reaction, both are transformed.

—Carl Jung

Sometimes there is a love-hate relationship between the business analyst and members of the solution team, starting with the project manager. Technologists who have a distrust of the users or consider them stupid tend to confer the same feelings to the business analyst who represents the user community to the solution team. Systems analysts and business analysts have a continuing battle over where the business analyst's job of defining business requirements ends and the systems analyst's job of defining system requirements begins, especially when the business analyst is a former systems analyst. And the project manager may find the business analyst's focus on solving the business problem irksome in light of deadline demands and politics. Developers who practice agile development methods look at the business analyst as an unnecessary extra layer of communication that could be eliminated with no loss of project or product integrity.

Business Analyst and Project Manager

"What is my relationship with the project manager?"

The project manager focuses on the project and the business analyst focuses on the product produced by the project. While the project manager

drives the project making sure the project stays on budget and schedule, the business analyst navigates making sure that the journey always leads to the right destination.

A high quality project requires a view of both the needs and attitudes of the business community who have the problem to be solved, and of the development community as they solve the problem. It is difficult for a single person to balance the pressures of both communities successfully. The alternative is to forge a Janus-like partnership between the business analyst and the project manager, where both are aiming at the same target and achieving the same vision. When the two are working in tandem, while serving their respective communities, the results are harmonious for the project and the organization.

Janus Relationship

In Roman mythology the god of gates and doors is Janus. In his capacity as god of doorways, he is also the god of beginnings and endings and was used by the ancient Romans to symbolize change and transitions, and the launch of new enterprises. Romans placed statues of Janus near doorways or created paintings of the god over front doors of houses so that he could look inward to protect the household and at the same time look outward to see what was coming. He is depicted with two faces, one looking forward and the other backward, literally giving him eyes in the back of his head.

The image of Janus looking forward to the new and backward at the old is an appropriate representation of the relationship between the project manager and the business analyst. Since an individual human does not possess the ability to view in two directions simultaneously, the project manager and business analyst form a relationship whereby they are working toward the same goal: The project manager looks inward toward the solution team to make sure the solution is being built correctly, while the business analyst looks outward to the customer and organization to ensure that the right solution is being built.

The project manager and business analyst have the same overall purpose: a high-quality product delivered by a successful project. The business analyst determines what must be done to successfully solve the business problem brought forth by the business community. The project manager determines how to efficiently solve the problem in a timely fashion. Acting in the role of doorkeeper, the project manager protects the project team from attention draining interruptions and the business analyst filters changes, issues, and problems that emanate from the business community. Those issues that the business analyst does not filter out are then filtered by the project manager based on schedule, budget, and feasibility.

When the two are in synchronization, the business analyst has no trouble representing the project manager's view to the customer and the project manager has a clear view of the product that will solve the customer's problem. They have the same partnership as the pilot and navigator: The business analyst provides the destination and the project manager drives the project. Through the business analyst, the project manager understands the scope of the product being delivered, its importance to the organization, risks to the business, and impact a given solution will have on other parts of the organization. Through the project manager, the business analyst understands the technical challenges and issues and is able to relay those challenges to the business in terms that allow the customer and stakeholders to make reasoned decisions about the project.

There is a strong implication in the Janus metaphor that the project manager and business analyst are of one mind. However, even with the close relationship required for success, the business analyst and project manager should provide a check and balance on each other. The business analyst makes sure the project manager does not sacrifice the solution to the exigencies of schedule or budget. The project manager keeps the business analyst within the limits of technological feasibility and realistic project performance.

When working with technical project managers, it might be the business analyst's job to forge this Janus relationship. The business analyst may have better developed interpersonal and communication skills by the nature of the job. That is not a knock against the technical project manager who deals more with technical issues and code; it is simply a statement of role differentiation. In the end, it is best if it is a mutual merging, although it doesn't matter who initiates the partnership, as long as the relationship exists for the duration of the project.

Vive la Différence

There is a difference in the makeup of successful project managers and equally successful business analysts. While most business analysts and project managers struggle with the line of demarcation between their respective arenas, the successful teams are very clear in the roles and responsibilities of both disciplines. The business community and the solution team are clear who to refer to in every situation.

The first and perhaps primary difference between the IT project manager and the business analyst is in their points of view of the project. The project manager views the project as their primary work. They bring the project in on time, within budget, while delivering everything promised for that time and budget. They see the world in the framework of that project, with its finite start and finite end. The project represents a unique, specific, and definable piece of work.

The business analyst, on the other hand, views the project as one of a series of ongoing changes to the organization, one step in the overall organizational improvement journey. A project solves a single problem, a small part of a larger business process that will outlive every project. The business analyst, in other words, looks at a much bigger organizational picture than does the project manager. The project manager is responsible for the success of the project; the business analyst for the success of the project's product.

Now, this is not to demean or lessen the importance of the project manager. The function of project management requires the focused attention that the project manager brings to the project. The success of the project is diluted, if not threatened, when the project manager starts diverting their attention to aspects outside the project other than those technical issues that impinge on other projects or the overall technical architecture of the organization.

''What is the best way for a business analyst to work with the project manager, especially if the project manager isn't really doing a good job?''

Building a Strong Relationship

The project manager may view the business analyst as a subordinate or one of the solution team. The business analyst has to establish the level of equality necessary to get balance in the project. This is easier when the business analyst has defined the product scope and prepared the enterprise analysis decision papers, and then the project manager comes on board when the charter is signed. The business analyst already has history with the business problem.

When both parties come onto the project at the same time, or as is most likely, the project manager precedes the business analyst on the project, establishing that equality is more difficult.

Here are some tips to help establish the equal relationship with the project manager:

- Keep your focus on the problem and solution.
- Keep a constant flow of communication going with the project manager.
- Express the feelings and opinions of the business community and the problem owner rather than your own.
- Remember and respect the project manager's commitment to the schedule and budget—it is his or her job.
- Remember and respect the project manager's responsibilities to the team and the technical solution.
- Keep the project manager in the loop for all interchanges with the solution team and all negotiations with the business.
- Stay out of the project politics.

"We work primarily on small projects and my boss expects me to take over as project manager after I finish defining the requirements. How can you do that?"

Playing Both Roles

If you chase two rabbits, both will escape.

—Proverb

Since none of us can really be Janus and have two faces facing in opposite directions, we are going to find it difficult to be both a professional business analyst and a professional project manager at the same time. Even shifting back and forth is difficult—unless you are schizophrenic.

You may find yourself in the position of defining the business problem to be solved and creating a business case or project charter. And when the project is approved, you are assigned as the project manager. This makes sense from a business perspective since the business analyst at that point probably knows more about the situation than anyone else, and has already established a good relationship with the business. From the reports of hundreds of business analysts, moving from business analyst to project manager is a normal progression on a smaller project in the organization.

A manager of business analysts told me, "The business analyst picks up the role of project management once we analyze the problem, define the solution, and select a technology. For most of our smaller projects we tend to have the BA also be the project manager during delivery of the solution." A great many of the business analysts I talk to also assume project manager roles. In some organizations it is part of the process: In the role of business analyst a person defines the project, and then assumes the role of project manager to execute it:

> The pure business analyst's role has become diluted. It used to be the business analyst who talked to users, ironing out the details of what they wanted and balancing that wish list against what an IT system could economically or practically deliver. It was also the business analyst who ensured that users developed business processes to support the software. The actual task of writing the software (or installing and configuring packaged software) fell to a project manager and his team of software developers. But those were the good old days. Today, companies commonly ask IT managers to assume the business analyst role in addition to their duties as project manager. This is especially true in small projects, those taking between 100 and 500 person-hours to complete.[1]

Checks and Balances

When confronted by a change to the specifications, the project manager should ask, "Will this change cause us to miss our deadline or break our budget?" The business analyst should ask, "Does this change help us to better solve the problem?"

The relationship you have with the business community and your knowledge of the business provide the project manager with needed perspective to embrace the entire project scope while still focusing on developing the solution. The focus you maintain on the quality of the solution provides a check and balance to the project manager's focus on schedule and resources, especially when there is a temptation to cut quality to achieve an imposed deadline. You can better relate project status and problems to the business community in terms that they can understand and deal with, relieving the project manager of the sometimes unsavory task of reporting bad news to the customer or having to negotiate for more time and/or money.

The business analyst and project manager perform checks and balances on each other as well as on the business community and the solution team. The business analyst checks on the quality of the product as it is being developed and relays to the project manager the feedback from the business community. The project manager keeps the business analyst and, by extension, the business community, honest as regards budget, schedule, and politics.

When the project manager manages change management he will be biased to accept or reject change based on the impact on the project schedule rather than the impact on the overall problem and/or solution. When the business analyst manages change, he or she will only focus on how it affects the solution without regard to impact to project schedule. Neither stance is correct; both stances are necessary.

Business Analyst and Systems Analyst

The systems analyst may be considered the business analyst on the technical or solution side, or the business analyst may be the systems analyst on the problem side. Both professions have similar duties. Both require analysis, both produce solutions, and both work for a project manager during the solution life cycle. Each produces a different outcome: The business analyst defines what is to be done and the systems analyst defines how it is to be done.

> *Design is the all-too-brief period during an application's evolution when we consider what we want to create and speculate about how*

we want to do it but are not yet fully committed to any particular approach to the problem.

—Payson Hall

Recalling the unofficial historical background of the business analyst described earlier, the systems analyst performed the business analyst role before the position of business analyst came into existence. A great many of the business analysts I talk to or work with from the IT community were systems analysts prior to their transition into the business analyst position. For some, the transition was smooth and effortless, while others continued to struggle, unsure of how to play the new role and how it differs from what they were doing.

Position

The systems analyst is the technologist who defines *how* the solution is going to be implemented. Systems analysts have as many names in organizations as business analysts. In a German bank they are called technical analysts. In an American bank they are called business technical analysts. Technical lead, functional analyst, system designer, programmer analyst, computer technical analyst, and even system architect are all titles I have heard applied to the role of the person who defines how the solution is going to be implemented. There are times when the systems analyst is a team leader directing or coaching the developers in the creation of the solution; and other times the systems analyst is an independent entity providing direction and technical advice much like the architect during the building of an edifice. The systems analyst may double up and perform the role of project manager or programmer. And, sometimes, the systems analyst and the business analyst are the same person.

Here is a remark from a business analyst who made the transition from systems analyst to business analyst: "All my life I've been paid to deliver a task outcome, a deliverable. This [being a business analyst] is more about trying to stand back and identify who can help you; who can you work collaboratively with."[2]

Differences

A business analyst is customer-facing and spends their time in the problem domain, gathering information from the business community, analyzing the information, diagramming the information, and producing the business solution. The systems analyst is system-facing and spends their time in the solution domain analyzing the requirements, diagramming the solution, and producing the system design and system specifications.
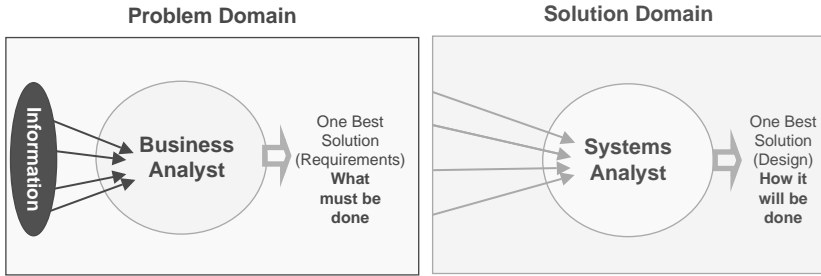
FIGURE 6.1 Difference between the Business Analyst and the Systems Analyst

As shown in Figure 6.1, the business analyst and the systems analyst both produce the one best solution from a problem statement. The business analyst works from the information elicited during the requirements definition process based on the defined business problem. The business analyst analyzes that information to produce solutions which are then further analyzed until the business analyst, in conjunction with the product stakeholders, derives the one best solution: what must be done by the business to solve the problem. That solution, or at least the part of the solution that is concerned with computer software and/or hardware, is represented as a set of requirements called the solution document.

The systems analyst receives the problem in the form of the solution document. The systems analyst's job is to determine how to implement the solution using computer technology. Because there are also a number of valid solutions that the systems analyst can implement to solve the problem, the systems analyst makes technical and business trade-offs to determine the one best solution, which the systems analyst documents in the system design and/or system requirements. This design takes into consideration the technical environment including hardware, interrelating software, networks, databases, and all other aspects of the technical solution.

Working with the Systems Analyst

Because many business analysts cycled through the systems analyst position sometime before becoming a business analyst, there is a love or hate relationship between the two disciplines. It is natural for a former systems analyst to write requirements that are more technical in nature and that have elements of the technical solution in them. And it is just as natural for the systems analyst on the solutions team to resist such attempts to do his job. Some current systems analysts view business analysts who once were systems analysts as lacking the technical skills to continue in that job. On the

other hand, some systems analysts welcome the presence of a technically savvy business analyst. It makes their job easier.

Trade-Offs

During the creation of the design and defining how the solution is going to be implemented, there are trade-offs that the systems analyst needs to make: trading off security for convenience, design elegance for deadline compliance, performance for flexibility, and so forth. There are a great number of decisions to be made during systems analysis and design. Many of the decisions affect the business community. This is where you establish your good working relationship with the systems analyst. You provide the information that helps the systems analyst make those decisions, and take the results of the decisions back to the product stakeholders for review. Normally, the product stakeholders have no veto power in technology decisions such as normalizing a database, or selecting an encryption level for data transfer, and these decisions are not brought to the business's attention. Even with those decisions that affect the users, such as when the users want a feature that does not comply with corporate standards, the user community has no veto power. This is when you step in and, in your diplomat role, tactfully suggest the changes.

When you run interference with the business community for the systems analyst during the design phase of solution development, you take a load off the systems analyst's plate and establish a good working relationship.

Getting Together

There is no rule that requires the business analyst and systems analyst to communicate only through documents. Documents tend to create boundaries between processes, departments, and people. Rather than focus on creating a document that the systems analysts will be able to read and use, increase direct contact with the systems analysts, getting them involved early in the problem-solving effort to advise on feasibility and overall technical approach, and continue to meet with them throughout the solution effort. Work to erase those imaginary boundaries that mostly exist in academic descriptions of software development.

Transitioning to Business Analyst from Systems Analyst

"I've been a systems analyst for over five years. How do I transition to my new job as business analyst?"

The most difficult part of the transition from systems analyst to business analyst is to suppress your tendency to start focusing on how the

problem is going to be solved: the technologies, the design, and even the code. I can remember many times writing code in my head while I listened to a businessperson describe what was wanted. I visualized the problem not in terms of what the business was going to do and what it was going to get, but in terms of the code that solved the problem. Without a concerted effort to break the habit, the tendency to jump to technical solutions before understanding the business problem can stay with you for years and slow up or even sidetrack your progress from systems analyst to business analyst.

As much as you can, step away from the computer and technology. View the problem from the business perspective: what the process workers have to do. What is going to have to change in their world to solve this problem? It is so tempting to design a solution and then assume that the business community is so happy getting the problem solved that they do not care about the impact on their work. They do care. And that is the reason a business analyst is around—to help determine the best solution for them.

Trying to Do All Jobs

In Appendix D there is a table of activities and tasks for the three roles. As can be seen, the activities and tasks are pretty much the same for all three. This overlapping of terminology is what brings about much of the confusion among those playing the three roles. The difference is in the focus for each. The project manager focuses their tasks and activities on the project. The system analyst focuses their tasks and activities on the technical aspects of the solution implementation. The business analyst focuses their tasks and activities on the solution to the business problem as understood by the product stakeholders.

In those not-so-rare circumstances when you have to perform one or more of the roles, hopefully sequentially, on—even more hopefully—very small projects, you act as a one-man band.

"I have to do everything from defining the requirements to coding and testing. How can I effectively be a one-man band?"

You can't.

As mentioned before, there is much to be gained by having two different people take on the roles of project manager and business analyst. There is more to be gained by having all three of the primary roles on the solution team played by different people. Each role has a different focus: The business analyst focuses on the product; the systems analyst focuses on the

technical aspects, and the project manager focuses on the overall project. As I mentioned in an article on wearing multiple hats:

> The problem we have when assuming multiple roles of project manager, business analyst, systems analyst, designer, and programmer is the inherent conflict between the goals of the roles. The project manager has to focus on getting the project done on time and with the assigned resources even if it might mean cutting back on the solution. The designer is after the most efficient way of solving the problem. The business analyst is customer-facing and wants to keep the customer happy without regard to time or resources. You can expect these roles to conflict when only one person is responsible for doing all the jobs. The project manager keeps the designer from getting analysis paralysis; the designer ensures that the solution is feasible; the business analyst reminds the project team that the results of the project must solve the customer's business problem. And so forth. When we don all hats at the same time we have to internally resolve the built-in conflicts, which lead to indecision, sleepless nights and erroneous actions. The checks and balances inherent in the process are better served with independent advocacy.[3]

As a young project manager on one of my first projects, I still tended to take on technical tasks and engage in technical discussions, trying to help out in all functions and activities. I would do requirements, write code, draw designs, and so forth. I got into the habit of responding to questions with the phrase "wearing my system analyst (or project manager or requirements) hat I'd . . . " One morning I came into my office and on top of the file cabinet were a series of several different colored baseball caps, each with a different insignia: "PM," "RQ," "SA," PG," and so forth, and a note from my team that said, "From now on wear the appropriate hat." It was meant as a joke and was intended to end there. Being a nerd I carried it forward, fashioning a hook out of a coat hanger to affix to my belt between my pagers where I could hang the hats. Then throughout the rest of the project when asked a question or starting a meeting, I'd switch to the appropriate hat. While it was meant as a running joke to begin with, I found that when I switched hats I was able to shift focus from one role to another. That shift helped me play the required role better. The project was reasonably successful, but management would not allow me to continue wearing the hats in successive projects so they ended up in a box somewhere.

Do whatever you can to separate the roles you have to play as project manager if, in fact, you must play the different roles. It is better to follow the PMBOK advice to just be project manager and nothing but. Assuming in these tight times that is not possible, differentiate the roles

as best you can. If you have to create the solution document, then complete it by noon, place it in a folder, and put a Post-It note on it addressed to you, as the designer. Then go to a long lunch away from your desk or office, come back as the designer and receive the solution document as though from someone else. Read it with the same critical eye you would if it had actually come from a different person. If you have been designing all afternoon and have a project manager status meeting to attend in order to report to upper-level management, take a long break from the design effort and do something different to get yourself into the role of project manager. In other words, try to separate the roles you are playing as much as possible. For example:

- Instead of grouping all the material pertaining to a project into one manila folder, or one directory, separate the material based on the different roles you are playing.
- Designate one side of your desk for one role and the other for the other role (assuming you only have two); do the same with file drawers. That gives you a chance to break from one role to the other when you change sides of the desk.
- Never have a meeting where you are required to play multiple roles. When multiple roles are called for have separate meetings with a break in between to allow you to shift out of the previous role and into the next.
- Have different hats to wear for each different role on the project and take the time to change hats when changing roles. It allows time to shift from right brain to left or from manager to worker.
- Do not forget to shift to the role of significant other, parent, jolly good fellow, game player, dreamer, or whatever when you are not in the office.

The concept is to see your work as others see it, with objectivity and detachment, and to make sure the audience knows which role you are playing at the time. Review your work and your role as you would review other people's work and their performance in their role.

Business Analyst and the Rest of the Solution Team

Other than in agile development methods, the business analyst typically does not relate directly to the solution team. Most communication with the team is done through the project manager. However, the more facile the information flow is between the business analyst and the systems designers, architects, database administrators, and the rest of solution team, the

more likely they will consult the business analyst on technical matters that may affect the business. This is not to say that such communication will be easy.

> *The trouble with programmers is that you can never tell what a programmer is doing until it's too late.*
> —Seymour Cray

The solution team consists of all the people who will come together to produce the solution. Since most of a business analyst's work is with computer-related solutions, the solution team is a team of developers, testers, database administrators, network administrators, systems analysts, programmers, systems architects, hardware analysts, and so forth.

Some business analysts perceive that their job is to support the IT project team and nothing more. Their involvement with the business community is solely to define what the project team needs and to occasionally be by the project manager's side during negotiations with the product stakeholders. As such, the business analyst becomes a part of the solution team.

The independent business analyst views the project team as an entity with the project manager as the spokesperson. The business analyst establishes communication paths between the project team and the business community and focuses on removing obstacles to communication.

You may find that the developers and the rest of the solution team are valuable allies in pursuit of a solution. Not only can they tell you what is technologically feasible so you can manage the business' expectations, they also have solutions and alternatives based on the latest technology. The developers can offer suggestions for parts of the solution based on existing systems and the team's abilities.

> *Everybody wants to build and nobody wants to do maintenance.*
> —Kurt Vonnegut

"Communication with the developers is not very satisfactory. They have no respect for what we do."

Dealing with Developers and Programmers

When I was a programmer in the early days of business computing, my job was to turn specifications derived from requirements of some sort into instructions that the computer understood, called machine language, by writing a machine-translatable code called a programming language. At that time, there was no concern among us programmers about whether the results of the code we wrote solved a business problem. We assumed that the

results would provide a measureable benefit to the organization because someone in the business told us we needed to do it. Besides, we were thoroughly enjoying the groundbreaking work we were doing. Just about every line of code we wrote was a new innovation never before accomplished. When on rare occasion we used the term *developer* it referenced all the parties engaged in developing software or computer systems, of which the programmer was a central role and we were proud of it. Nowadays there appears to be a clear difference between a developer and a programmer, although that difference may only be known to the technical side of the house. Dr. Steven Gordon defines the modern term developer this way: "A developer is actively engaged in the entire software development process and takes responsibility for the value delivered to the customer."

Clearly, there is an effect on your relationship with the solution team depending on whether the team is made up of developers or programmers. Since developers have a wider view of the overall problem and solution, they need more involvement with the outcome and, therefore, expect to be included in the evaluation of the solution. Programmers expect to be told what to program and assume that whatever they produce is of value to the organization.

Team Composition

The solution team may be a co-located group of developers working together as a self-managing team in an agile environment. It may be dispersed over several continents and working in silos communicating solely by e-mails and documents. The team may be a group of individuals who rarely speak outside work or, conversely, they may stay together after work for social events. You may encounter a work area with two developers sitting together at each workstation doing what is called pair programming. You may see the team spending hours drawing diagrams on whiteboards that they erase immediately. You may see a programmer staring at the ceiling for hours. You may see several developers playing foosball or air hockey. Regardless of the social, logistical, or technological make up of the team, it has one purpose only: to turn your definition of the solution to the business problem from a document into reality.

From the business analyst's perspective it is important to remember that the team is there to solve the problem regardless of composition or appearance, or even demeanor. And it is even more important to communicate this to the business. The business analyst is not a matchmaker and does not have to create a *Friends*-like atmosphere. However, the more the business analyst can relay to the process workers and business management about the difficulty of creating systems from nothing more than vaguely described issues, and the more the business analyst can work with the business community to

clearly define the issues, problems, and expected solutions, the more communication will flow between the problem and solution domains.

Buy, Not Shop

It is probably a male-female thing. When my wife and I met I told her I liked shopping. That was a plus in her evaluation of my character. After we were married, it turned out that we had different definitions of shopping. When I go shopping I determine what I want, go to the store that has that item, purchase it, and return home. When my wife goes shopping she may spend hours in the effort and purchase nothing. Shopping includes identifying the alternatives, evaluating them, comparing them when possible, discussing the alternatives with a shopping companion or expert, and, when a conclusion is reached, purchasing the item. What we determined is that my wife goes shopping and I go buying.

Developers expect the business to be buying and not shopping. Developers expect the business community to know exactly what is needed and be able to specify that need exactly. When the stakeholders exhibit indecisive behavior or a desire to evaluate alternatives, the developers are convinced the users never know what they want. The developers expect the users and other stakeholders to know precisely what they want to solve the business problem. The users, on the other hand, expect that their problem will be solved by IT in the same way a doctor removes their pain when they vaguely describe their discomfort.

One option to resolving this buy–shop differential is to side with the developers and roll your eyes skyward whenever a change is requested, and mollify the developers with comments like, "Well, what can you expect? They're users." Or on the other hand, you could remind the developers that, after all, the system being built is something that the users are going to live with for a long time, perhaps their entire business lifetime, and therefore they should be accorded some latitude in making the final decision.

Another option is to work with the process workers before exposing a final or near-final decision on the solution and let them shop with you, so that you bring the buy decision to the developers. You can create solutions with the process workers and other product stakeholders with storyboards, prototypes, wire diagrams, scenarios, and so forth, so that the shopping is done. This is like shopping on the Internet, but buying in the store.

A third option is to adopt a more agile or iterative approach that involves the developers in the definition effort. The developers are brought in on the shopping game and in short iterations create alternatives that the stakeholders may evaluate and select, reject, or change. The developers then have the opportunity to create a finished product, and the stakeholders

are not forced to make a final decision on the solution until they have evaluated the real alternatives.

Ultimately, the business analyst acts like a shopper who turns a customer's vague desire for an item into a purchase decision, by taking a businessperson's unease with the current situation and defining specifically what needs to be done in language understandable to both the business and the developers. The business analyst then presents the buy order, the result of the examination of the alternatives, to the solution team.

As can be seen by the process described in the following parts of the book, a major objective of the process is the clear and concise definition of the problem and subsequent solution, even if the solution is not complete as in an agile approach.

Challenges

There are a number of challenges in dealing with the solution team; however, the more the business analyst can get the process workers and other stakeholders to the table with the developers, the better the solution will be.

Understanding the Bigger Picture   In my early days of programming, I wrote a lot of code without knowing how the results were used. I have no idea even today whether the company benefited from the programs I wrote or not. I assume they did because they paid me. To me it did not matter. The program executed correctly. The numbers added up accurately. The bugs were fixed. There were no complaints from the users. I was more interested in an elegant programming solution than an effective business solution. Most developers today still are more interested in solving the technology problem than the business problem. Here is the explanation of one developer who was commenting on his analysis: "In terms of analytic ability, I do ten times more work on analyzing the efficiency of a sort algorithm as I did on analyzing requirements."

One of the aspects of dealing with the development team is convincing them that software development is not only about writing code. The software must have a purpose and it is not just passing the functional test or calling the right method. The software must integrate with an entire solution that has value to the business. For many technologists, the business is all politics. They expect the users, or management, or the product owner to have all the answers in terms of resolving business conflicts and ensuring the solution is correct. All they want to do is code. However, as Frank Hayes points out, that is not a good career move nowadays:

> Pure techies don't see the point in understanding and navigating the labyrinth of office politics—why it's important to know who's got

clout, whose users really matter and which projects have real back-
ing. It's too pointlessly Machiavellian to them. That's why they will
probably be blindsided by the next big purge that flushes them
away. And they don't get why it's important to understand the com-
pany they work for and the industry their company does business
in. So they will make business-clueless mistakes. They will meet the
specs for the Sarbanes-Oxley project, but it won't pass the audit.
They will hold off rolling out the new sales application because
some widget isn't working quite right, even though revenue is be-
ing lost. They will solve technical problems brilliantly but fail to fill
real business needs.[4]

To overcome this developer tendency, the solution should be presented
to the solution team in terms that define the benefits to the business and why
process workers need to have the solution implemented in a specific way.
Don't assume the solution team won't understand or are not interested in
anything except the technical aspects. Do assume that the members of the
solution team are just as interested in a positive solution to the business
problem as you are. When you present the solution or answers to their ques-
tions during solution development, keep the politics out. The solution team
does not have to know about conflicts between business units over the solu-
tion, or interim decisions of management that are reversed, or the difficulty
you have in obtaining information from a group of product stakeholders.

No Problem Syndrome   Gerald Weinberg discusses the No Problem Syn-
drome (which he calls NPS) in his book, *Becoming a Technical Leader.*[5] He
describes the situation of computer professionals who come up with a solu-
tion without fully understanding the problem, much less the requirements.
They respond to any business request with "No problem" instead of some-
thing like, "I'm not really sure what you are talking about, but we can try a
few things until we have it solved," or "That appears to add a lot of com-
plexity to the system which may make the system harder to use." He posits
that this statement of "No problem" is a warning sign that something is
wrong or going to be wrong. While the business manager hears "This
should be easy and I can do it in no time for very little money," the devel-
oper is actually saying, "I haven't a clue if I can do what you're asking. We'll
just wing it until we have it done." As Gerry Weinberg observes:

I used to think that computers emit some nerve-damaging high fre-
quency sounds, because NPS seems to affect a large percentage of
computer professionals. Whenever they hear the words 'problem'
and 'computer' in the same sentence they launch into a diatribe.
And it always starts with the words 'no problem.'[6]

This is an insidious and difficult situation. The technician states that he has the solution to the problem well in hand, and therefore it is no longer a problem. The businessperson has no reason to disbelieve the technician, especially after the technician describes the solution in either vague terms or in such technical terms that it sounds as though he knows what he is talking about. When the technician delivers the solution, and it does not quite solve the problem, the technician will challenge the businesspeople for not providing him with enough data, or spin the solution so that the businesspeople feel as though they should use it as is and perhaps the problem they thought they had is not really the real problem at all.

When you sense that you are faced with a no problem syndrome response, ask for clarification about what it is that the developer is going to do. Usually the developer will have no compunction about explaining it to you. When the explanation is more technical than you can understand, ask for simpler explanations until you do understand. While listening to the explanation ask questions that will encourage further thought and consideration on the developer's part. Should it become clear that the developer has jumped to a solution, ask how the solution accommodates those areas that concern you. Eventually, you and the developer will come to a mutual understanding of the problem to be solved and what the developer is going to do. When the developer is reticent to explain the solution he has in mind, perhaps because he is afraid you will not understand, remind him that you have to explain what is going on to the business community, and you would like his help in doing so.

Talk to Me   The agile development community prefers communication with the business through a product owner or onsite customer. The developers do, however, limit the communication to only one representative of the business to lessen the political ramifications of change. Outside agile methods, programmers tend to be more reticent. As Alistair Cockburn, author of one of the seminal books in the agile lexicon, *Agile Software Development*, states, based on his years of working with developers that "it's generally a convenient fiction to say that programmers want to talk to the customers and understand what's in the customer's mind, in many cases it simply isn't true." He mentions that as he visits organizations, "many programmers really don't want to talk to the business or anyone; they just want to sit and code."[7]

One reason for the lack of communication may be inherent in the personality of the typical programmer, who tends to be introverted. Another is due to management, which sometimes goes to great lengths to keep the developers away from the business because of long-standing stereotypical images of the nerd programmer. Last, it is partly because many technologists really do not want businesspeople to know what they are doing. "In fact, many technology 'experts' go out of their way to keep others in the dark, almost reveling in their personal reputation as a tech wizard."[8]

When you wish to work with the technicians and become more fluent in the technology that the technicians are using, spend more time listening and asking questions. When you are truly interested in what they are doing and how they are doing it, the technicians will talk with you.

You may find yourself between upper-level management and the solution team. There is almost as great a gulf between the suits (as the technicians call them) and the nerds (as one of the more printable names upper-level management calls the technicians) as between the sales and marketing types and engineers.

Here are some examples of what CIOs think when the subject of developers comes up:

- "IT to them is like electricity: They need it, but they don't appreciate it," says the CIO. "Having the 'prima donna' developers' attitude that the organization exists to provide them with some intellectual stimulation is not what the executives want to hear or feel!"
- "Some developers think it's all about their code. They fail to understand the mission of the business and that they don't drive it, they support it," notes the IT manager. "Their work is often not mission critical or urgent."[9]

You need to work on both sides to ameliorate the animosity and close the gap. It is as likely you can get the technicians to wear suits and talk balance sheets as getting the VP of finance to wear sandals and play *Doom* for days. At least you can help get them understand, through your role as educator, that both parties have a place at the table.

As a business analyst representing the business community to the solution team, you may not be considered a direct member of the team. However, the relationship you have with the other members of the solution team is critically important. Disabuse yourself of the stereotypical notion that all programmers do is write code and they want no involvement with any other part of the solution. Developers today have a wider interest in the business, want to be involved with the users, and thrive on the feedback the users provide to their deliverables.

Bottom Line

Dealing with developers may be easier when you understand what developers expect and how they can either comply with that expectation or manage it. Many developers assume that they are dealing with clueless people who do not know what they want and are not going to like it when they get it anyway. Many may believe that they generally know more about what is really going on than the users do. The users are using IT's computers and

running IT's software. Without the developers, the users do not have a chance. Here's a quote from one such developer, "We never get requirements from customers. We only get arcane requests and statements of needs. As far as I can tell, a need is just their vision of a requirement. We still have to translate those needs into geek-speak. No way around that."

There actually is a way around that. The business analyst can provide the translation from arcane requests to something that may not be quite geek-speak, but will suffice to allow the users to confirm and the developers to understand. Eventually the business analyst can serve an even greater good: educating both the process workers and the developers so that they can indeed communicate directly and successfully. You don't have to adopt or employ agile development methods to increase the communication between the developers and business community. Simply include developers as much as possible in the overall solution process to increase your chances of a successful solution implementation.

Dealing with the solution team may be hard enough and may challenge every communication skill you have. However, that is only half, or even less than half, of the communications you do on a daily basis as a business analyst. The business analyst actually should be spending more of their time with the business community, both the product stakeholders and those who may be the stakeholders for the next product. The business analyst may be a negotiator, mediator, facilitator, coach, confessor, and, of course, blame-taker for the business community. We look at the various demands of communication with process workers, business managers, and upper-level management in the next chapter.

Notes

1. Steve Gilbert, "Wearing Two Hats: Analyst-Managers for Small Software Projects," *IT Pro* 6 (4) (July/August 2004): 34–39.

2. Beverley Head, "Bridges over Troubled Waters," *CIO* (August 8, 2007), www.cio.com.au/index.php/id;889859495;pp;1.

3. Steve Blais, "The Hat Trick," www.AllPm.com (April 9, 2007).

4. Frank Hayes, "Survival Skills," *ComputerWorld* (September 5, 2005), available from http://www.computerworld.com/s/article/104355/Survival_Skills.

5. Gerald Weinberg, *Becoming a Technical Leader* (New York: Dorsett House, 1986).

6. Ibid., p. 67.

7. Interview with Alistair Cockburn.

8. Jim Rapoza, "Tech Ignorance Isn't Bliss," *eWeek* (November 3, 2008), 49.

9. Thomas Wailgum, "8 Reasons Why CIOs Think Their Application Developers Are Clueless," *CIO* (September 3, 2008), www.cio.com.

The Business Analyst and the Business Community

If you want to teach people a new way of thinking, don't bother to teach them. Instead, give them a tool, the use of which will lead to a new way of thinking.

—Buckminster Fuller

The product stakeholders are the part of the business community who comprise the target audience for the business analyst. These are individuals who experience the change first-hand, whether affected by the problem or impacted by the solution, and have the information that the business analyst uses to define the problem and come up with the solution. The business analyst's relationship with the product stakeholders can often make or break the project. The relationship between the stakeholders is not one of continually providing satisfaction or making them happy. It's about managing expectations. The primary approach to managing your stakeholders' expectations is to include them in the overall process, to let them see, understand, and be part of the systems approach to solving their problem.

According to the Project Management Institute's *Guide to the Project Management Body of Knowledge*, stakeholders are "persons of organizations (e.g., customers, sponsors, the performing organization, or the public) who are actively involved in the project or whose interests may be positively or negatively affected by the performance or completion of the project."[1] Or as a friend of mine from the United Kingdom once said, "The stakeholder is anyone who might complain about the project when it is over." In systems projects, this includes the development

team, the DBA(s), quality assurance and the testers, portfolio managers or review boards, upper-level management, the project manager and business analyst, as well as internal and external customers, users, and the executive decision maker. All of these hold a stake in the outcome of the project, either actively or passively.

Constituents and Constituencies

"Biggest issue I see is a lack of common language among all the different business parties."

Business analysts deal with constituencies. A constituency, for our purposes, defines a functional area of the organization. It is a group of process workers who work on the same business process to achieve a single process goal; a generally cohesive group of people who will act with or react to the business process. Some examples include accounts receivable, the collections department, compliance, the help desk, the CEO, regulatory agencies that deal directly with the organization (e.g., the SEC or FDA), vendors, customers, and marketing. They get the same information, perform the same activities, have similar responsibilities, and have a unified goal in terms of its functionality.

By using this definition we include all forms and levels of business community personnel who might have information that impacts the definition and development of the product. Within our constituency, there is a central group of constituents: the product stakeholders. Note that our term *constituent* is not limited to human beings. A constituent, much like an actor in a use case, can be anyone or anything that has a requirement of the product to be built. In other words, another system may require data produced in a certain format. That system is as much a constituent represented by the business analyst as any person using the system.

Business Analysts and Upper-Level Management

The business analyst has an advisory relationship with upper-level management to provide objective counsel for the projects and products undertaken by the organization. Management is informed about the status of the project by the project manager and IT management. What they may not know about is whether the project still needs to continue in light of changing problems or objectives, the readiness or resistance of the user community as regards the eventual solution, and other business or problem-oriented aspects of the overall project.

The business analyst provides that information to upper-level management.

> *Management requires holistic thinking, heuristic judgment and intuition based upon experience.*
>
> —Tom DeMarco, PeopleWare

The term *upper-level management* refers to any member of the organizational hierarchy in a position of authority or influence over the project or the choice of solution. The individuals occupying this category are usually not directly involved with the problem-solving effort other than getting reports on progress and rendering decisions. In the following sections we discuss a few of the general categories of upper-level management with whom the business analyst typically works.

IT Management

The business analyst usually does not have much to do with IT management beyond the project manager. The relationship with IT is through the project. However, there are instances where the business analyst must be wary. One is when IT management is concerned with saving face (or budget) and orders the business analyst to "find out what they want so we do not deliver the wrong thing," rather than "determine the real business problem so we deliver something of value to the business."

IT is always committed to delivering a system that is of the highest quality possible. This commitment is tempered by deadlines and budgets. The business makes the decision to accept the system at less quality to meet a business deadline or because of cost restrictions.

You are viewed by IT management as the representative of the business. As such you are expected to advise IT management directly or through the project manager of project changes and issues affecting the business. In their eyes, you are playing the role of liaison or emissary (as discussed in Chapter 5). You may be asked to relate unpleasant news to the business on behalf of IT, and you may be asked to negotiate aspects of the solution representing IT. This is an opportunity to use your communication skills and gain much political credit, as well as help move the project to a successful conclusion.

Executive Decision Maker

The executive decision maker is the one with the authority over both the product and the project. They can cancel the project, find additional funds to make the product happen, and decide ultimately whether the problem is worth solving, whether or not there is a cost benefit for solving it. Both the

solution side and the problem side of the project eventually report up to the executive decision maker.

Since the executive decision maker is typically on the business side, the business analyst will more likely have a relationship with the executive decision maker than with IT management. In many cases the executive decision maker is also the problem owner. In this case, the business analyst will have a close relationship with them. The important factor in the relationship is to make sure the executive decision maker is fully informed about the business problem and the solution. When the executive decision maker recognizes the benefits of solving the problem and the risks of not solving it, they can be a powerful supporter when obstacles crop up along the path to the solution.

Business Management

It is a tricky situation when you are dealing with both the users of the system and their managers together. However, separating the process workers from their managers creates an even worse situation. I have had the unfortunate experience of trying to implement a system at the behest of user management when the users had no say in what the system did or how it operated. Basically, management decided a new system was in order and the users, being employees, would like it or they could leave. It is not a pretty picture. On the other hand, when users get full sway to change whatever they want and management is less than aware of the changes, other problems crop up: over-budget, behind schedule, systems that are nice to use and functional, but do not solve management's basic problems—it is not a pretty picture, either.

The business manager is many times the problem owner, or very close to the problem owner, so the business manager will be one of the product stakeholders and involved in the problem and solution definitions. Cultivate a strong relationship with the business manager as well as with the problem owner. The business manager may be your point of contact to direct you to the stakeholders to talk to during the elicitation and evaluation. Being everyone's boss in the problem domain, the business manager may cut through resistance that users might have to attend your endless elicitation meetings or review sessions. When managing expectations, you should start with the business manager's expectations first.

The primary information the business analyst supplies to upper-level management is about the product and the product status. The business analyst is not authorized to speak of the *project* status and should be wary of getting into those kinds of conversations. The customer, problem owner, or other business role may call upon the business analyst as the primary communicator in the business to make a presentation to upper-level

management, because the business analyst represented upper-level management in defining the business case. They may ask, "What's really going on with the project?" They may not want to talk to the project manager, or to anyone in IT, or may not have access to the solution team at all. In any case, tactfully decline to say anything about deadlines, positive or negative, or any issues the project is having, positive or negative. This is the purview of the project manager.

Staying in the Loop

"In meetings between stakeholders and upper-level management, the information derived doesn't flow down to the business analysts."

Make sure that when you begin an engagement for any part of the business that you establish a ground rule: You cannot solve the problem without all the information. Don't be concerned with discussions about the problem domain when you are not present. This is natural. After a meeting in which the process workers described the problem domain to you, they typically get together to compare notes and fill in each other's information. You get the results of that conversation in your follow-up interviews or meetings. You might be more concerned about decisions that are made about the solution in your absence. Legitimately, the only reason for your presence is for advice and counsel, and perhaps to record the results. Decisions the business is making concern the process they are going to have to live with long after you have gone to another problem to solve. They may want to have the unfettered freedom to discuss things away from all influence of IT, and, remember, you are viewed as representing IT. As long as you get the results of the decision and some rationale, being included in the full flow of the discussion is not necessary.

Product Stakeholders

The product stakeholder is our term for a subset of the overall stakeholder population that is of specific interest to the business analyst. The product stakeholder is one who is affected by the problem or impacted by the solution. That definition narrows down the number of people included in our information gathering plan to a manageable number and focuses our elicitation efforts so we are not spinning our wheels in wasted interviews and meetings.

The product stakeholder is interested in the benefits he or she will receive from the product (what is in it for me?), any other impacts the product will have on them (do I have to change what I have been doing to gain the

benefits?), and what their contribution to the project will be (can you just give me what I want without all these meetings?).

It is not as easy as it appears to identify all the stakeholders. Leaving a constituency out of the investigation may be political suicide, but even worse, it may mean that valuable information is not acquired which may affect the solution. It is at the minimum bad form to deliver the solution to the user community only to find members of the user community were not consulted for their input.

The primary product stakeholder is the problem owner, without which there is no problem. I discuss the problem owner in the following section. I also discuss some of the positions that may be problem owners or at least influential in the definition of the problem or solution. The process for determining the rest of the product stakeholders is discussed in Chapter 9.

The Problem Owner

The problem owner is generally the center of the business analyst's efforts on the business side, much as the project manager is the center on the solution side. The problem owner is usually easy to pinpoint— they are the person who brought the problem to the attention of the powers-that-be to start the problem-solving effort. The problem owner does not have to be personally affected by the problem, but does need to be aware that the problem exists. The problem owner tends to be the primary point of contact (PoC) for the business analyst in the business community.

Here is an example of how it might work:

Charley works in accounts payable. He enters accounts payable vouchers. He has been working at this job for six years and is the senior person in the department. Each workday Charley comes to his 3270 work station where there is a pile of accounts payable vouchers on the left-hand side of his desk. As he enters the vouchers into the accounts payable system, he places the completed voucher on the right-hand side of his desk. At the end of the day the pile has moved from left to right and Charley can leave and go to happy hour.

Charley has a problem. He is not getting out of work at 4:30 and therefore is missing at least part of happy hour. Legitimate or not, this is a real problem to Charley. Charley does not have the authority to seek a solution for the problem. All he can do is bring it to the attention of Susan, his supervisor. Susan has her own problems. Her overtime budget is nearly 75 percent expended and it is only two months into the new fiscal year. She is going to have to go back for

more overtime money in a re-budgeting meeting. She sees there is a problem, and she does not have the authority to seek a solution for it. She talks to her manager, Mary.

Mary recognizes the problems that her minions have. She does have the authority to seek a solution. She calls IT and talks to someone who agrees to send a business analyst over to investigate. Mary is the problem owner. She represents those who are actually feeling the pain.

Although the process worker, Charley, may be having the real problem,—perhaps a poorly designed user interface—it is the manager, Mary, who can do something about it.

Problem owners have the following characteristics: They

- Know why this project is necessary.
- Have the authority to seek a solution to the problem.
- Are able to define or approve the real business problem.
- Are able to identify, create, or approve the vision.
- Are a product and project resource.
- Are usually the primary point of contact for the business analyst or delegates a primary point of contact.
- Initiate the action to solve the problem.
- Generally need the solution for personal or political reasons.

The problem owner is a role. The problem owner does not necessarily have the authority to approve the project or allocate the budget, but he could. For example at one company the problem owner was the CEO. It was the CEO who wanted the old online system replaced by a new in-house system. The staff did not really care. They actually resisted the idea since it meant going through the painful process of learning a new way of doing things. Once the new system was up and running and the CEO's problem was solved, but the enhancements, changes, and improvements were problems identified by various organizational levels in the company who became the problem owners for their changes.

A common statement from business analysts is "None of the managers want to be part of the project. The users keep telling us stuff and the managers keep overruling them, but the managers don't want to initiate or take responsibility for any of it."

When there is no problem owner, there is no problem, or at least not one worthy of solving. There are many instances where a manager perceives a problem on behalf of someone else who has no idea there is a problem. The manager disavows ownership and the purported problem owner has

no interest. In this case, there is no problem because no one cares—there is no problem owner.

Then, again, in our efforts as humans to foist problems onto someone else, one department might indicate that another department has a problem. For example, the forms from department A have not been correctly verified by department B. This means that department A has to review the forms when they are returned by department B and send them back again, sometimes without any changes. Department A wants department B's processes changed to resolve the problem. Department A does not want to own the problem and department B does not believe it has a problem. The net result is that there is no problem, and there will remain no problem until someone takes ownership of it. Department B will continue to send back the forms and the business analyst investigating the issue on behalf of department A will find they are in a bind between the two departments. Until the business analyst is able to convince one or the other department managers (or a higher authority) to own the problem, they will get nowhere. Of course a higher authority may be called in to resolve the situation, in which case the higher authority becomes the problem owner.

It is important to identify the person who really has the problem, or who owns up to having the problem. This is not always an easy task. In many cases, business analysts are dispatched to gather information from process workers based on a problem defined by upper-level management. Upper-level management perceives that the process workers need a new function. The process workers, however, are perfectly happy with the current functionality and when confronted by business analysts eliciting information to solve their problem, will often exhibit attitudes ranging from total disinterest to outright hostility or even sabotage.

🔍 Example

A manufacturing company had a system that collected all the safety and other test data and produced reports that were submitted to the government to gain approval to sell their product. The system was over 20 years old and was written in Fortran and assembly language. The users were perfectly happy with the system as it was. Upper-level management was concerned that the two people who were maintaining the system were getting on in years and would be retiring soon, leaving the company with no one to maintain the system. Without the system, the company could conceivably not be able to sell any new products. Clearly, the problem did not belong to the users. They had no issues with a system that they had been using successfully for years. The assigned business analysts encountered

resistance from the users that was supported by supervisory and union management. The users didn't have the time to spend telling the business analysts what the system did and how it produced the reports for the government. The users did not need a new system or the trauma of change that comes with it. The two technicians maintaining the system were similarly too busy keeping the system running with changes to report formats and modifications to regulations and had long since lost touch with the business aspects of what they were doing, being so involved with the programming and technology.

In this case, the specific problem owner was in the ranks of upper-level management and was not identified. The business analysts had orders from the top to define what the system did in such a way that the entire regulatory compliance system could be replaced by a new system written in Java or some other more modern language with more modern technology. The users of the old system could care less about getting a new modern system, preferring the system they were used to, and were totally uncooperative. And there was a hard deadline: the day the last of the maintainers left the company for a well-earned retirement in the Florida Keys.

If you were the business analyst in this situation, what would you do?

Other than business management there are several other positions or roles that tend to be the problem owner. Sometimes, management assigns a subject matter expert (SME) to be the problem owner. This is not always a good situation. Here are some other roles or positions that are better candidates for problem owner.

"Our product manager makes all the decisions about requirements for the work we do. He sometimes defines the requirements and they aren't always right. How is it supposed to happen?"

Product Manager

The product manager is in charge of one or more of a company's product lines and works with marketing and sales to identify new products, and track the sales of existing products. Not all organizations have the position of product manager. The product manager interacts with business analysts and the solution life cycle when the products the company sells have software components as part of the product, or when the product manager needs the expertise of the business analyst for investigation or data analysis.

In most organizations, the product manager prioritizes the product feature set based on the product manager's understanding of the end user and the marketplace. The business analysts produce the solution document that defines what is necessary to implement that feature set. In the more successful product-oriented organizations, the product manager and business analysts work together. Each brings different perspectives to the table and together they arrive at a result that is better than either could have produced alone.

Generally you should treat the product manager as a problem owner because he does own a problem and you, as business analyst, are solving that problem: the introduction of new features into the existing product line, or modifications to products in the line, or an entirely new line for which new supporting systems are necessary.

Customer or Sponsor

Traditionally, we in IT define the customer or sponsor as the one paying for the product to be developed: the contracting officer in a government contract, for example, who may have no detailed knowledge of the work being done on the contract (that's the job of the contracting officer's technical representative—COTR) but whose signature is necessary to accept the results and get paid for the work. Even though the person in this role generally takes a hands-off attitude, the customer or sponsor may also be the problem owner.

There are a few issues with the concept of customer as IT typically views it:

- There is already and always a customer class in any organization: those who keep the organization running by purchasing or using the organization's products or services. Sometimes IT refers to these as external customers to distinguish them from IT's customers within the organization. This overloaded terminology can be quite challenging to the business analyst who may be dealing with both types of customers. Even worse, people in IT tend to think they are working for their "customer" or the internal customer when in reality all work for the organization is always done for the organization's customers, the ones who keep the organization in business.
- Many business analysts believe that their primary job is to make or keep the customer happy. The underlying rule is the old salesman's adage: the customer is always right. That may be valid in sales. However, it is not the business analyst's job to make the customer happy. It is the business analyst's job to solve the business problem.

- Referring to everyone on the business side as a customer blurs the real roles each person plays. One customer may be the problem owner; others may be accounts payable data enterers, supervisors, or managers. During elicitation, and more importantly, during solution definition, you need to focus on the roles each individual plays.

Try not to think of the business community as customers. Think of them more in the role that they play for the organization: process workers, users, managers, and so on. You want to establish a long-term relationship with the business community and thinking of them as customers does not lead to that relationship.

Product Owner

In Scrum and other agile development methods, a product owner represents the product stakeholders to the development team. The product owner is a communication conduit between the development team and its product stakeholders typically hailing from marketing.[2] In many instances the product owner and the business analyst are one in the same. When the product owner is a separate entity, the product owner can be considered by the business analyst to be the problem owner as well. The product owner defines the features, functions, and user stories to be implemented and the priority of implementation. As such, the product owner usually understands the problem and the various potential solutions to the problem. You will have a close relationship with the product owner throughout the Scrum project, sometimes representing the product owner to the solution team, and other times representing the product owner to the business community.

"We don't get good requirements because the SME is not an expert."

Subject Matter Experts

The Business Analysis Body of Knowledge (BABOK) defines an SME as "any individual with in-depth knowledge of a topic relevant to the business need or solution scope."[3]

Upper-level management seems to believe that there is an SME in every functional area. Usually the SME is designated as the one person—the manager or the worker—who has been there the longest, and therefore who must know everything about the process. The concept of the SME is generally based on a belief that there can be a single expert who knows everything about a complex process or situation. In real life this does not tend to be the case. In truth there is not one real expert, only one person

who knows everything about a particular part of the business. The knowledge is usually spread among many, a bit of knowledge here, a pertinent anecdote there, a best practice somewhere else. To complicate our search for the absolute truth about a particular area of the business or operations, each one of the subject matter experts has a different opinion of what the subject is, what really matters about the subject, and even the level of expertise necessary to be successful in their assigned subject. In other words, it is rare to find a subject matter expert who can answer all your questions and give you all the answers. Even with the best of SMEs, you still have to solve the problem yourself.

SMEs can be quite helpful and save a lot of time. They may also see solutions that you and others may not be able to see even when looking at the same information. They understand the history of how and why processes developed as they did. This history of the patterns, relationships, short cuts, and insights are not built into the business rules or policies and procedure manuals. They are only located in the experience of the SME and other process workers, and must be ferreted out through investigation.

Issues

Working with an SME can be a gratifying experience and the SME can make your job as a business analyst easier by providing all the information about a process from one source, which reduces the time you spend in investigation. However, as with everything else, the business analyst needs to apply analysis to the identification of the SME and the information the SME supplies. Here are some typical SME issues:

- Many times the SME is the *only* source of information, or at least it appears that way to the business. When this occurs and the SME is assigned to solve a problem, the solution becomes the SME's solution and no other solution is considered.
- The SME may have hidden agendas or political concerns and the business analyst must be on the lookout for them. The SME may want to ensure job security, raise her worth to the business, or forward her own political agenda.
- The SME may be the primary source, but not the only source. One of your responsibilities as business analyst is to discover or generate multiple solutions based on the input of many sources, only one of which is the SME. In questions of conflicting information, analyze all the information and do not accord correctness to the SME simply because he is the SME. The SME may feel that it is her duty as SME to provide answers to all questions, whether she knows the answer or not.

Example

I was gathering information for an insurance company for a large revision to one of their systems. The vice president sent me to talk with (we will call him) George. The vice president designated George as the SME in the existing process. I dutifully called George and said I wanted to talk to him about the new system and set up an appointment. George seemed very interested in talking to me so I figured it was a good sign. When I got to his office he asked as many questions of me as I did of him. He did not have a lot of answers. At some point, I asked how long he had been working with the system and he replied "not long." I blurted out that I was told he was considered the subject matter expert on the system. "Is that why you are here? Who told you I was the SME?" I told him that his boss said so and he laughed "I'm no SME—I barely know the system." George, it seemed, had most of the answers in meetings that the vice president attended and had done some work for the VP, which explained the VP's recommendation. George thought I was there to describe how the new system was going to affect his work. I managed to get the information I needed from other sources, with George's help actually, and we developed a good friendship throughout the development of the new system. However, he was not an SME.

The lesson here is to prepare an information-gathering plan to determine the information you need. Armed with that plan you can qualify your responder by first confirming with the problem owner or manager who identifies the SME that the person has the information you are seeking, and then requalify the person before the interview or during the interview introduction. While the designated SME may not have the expertise you are looking for, they may have good information. Knowing in advance what general knowledge the SME has and does not have makes the information gathering session more productive.

The Good, the Bad, and the SME

In general, an SME is a positive addition to the product stakeholders. It is your responsibility to maximize the benefit of having a designated SME by confirming the SME's knowledge and area of expertise and making sure that you do not depend solely on the SME for the solution. Table 7.1 compares the positive and negative aspects of the SME.

TABLE 7.1 SME Comparison

Positive Aspects of SMEs	Concerns about Using SMEs
Save time by providing a single source of information on the problem domain.	May rely on SME too much so solution becomes the SME's.
Quick validation for all information gathered in the problem domain from all sources.	SME may answer questions whether or not the SME knows answer.
Because of their familiarity with the problem domain they can come up with better solutions than we can.	SMEs may have hidden agendas.
They know shortcuts and work-arounds unknown to anyone else.	The designated SME may not be a SME.
They are more familiar with more features of any operational system in the problem domain.	The SME may be a political appointee.
They will be able to identify reuse of business processes and activities.	The real SME may be the expert and not want to be involved in the process.

Process Workers

User, n. The word computer professionals use when they mean idiot.
—Dave Barry

The process worker knows how the business process works at the detail level; at least the details of his part of the process. Most of the information the business analyst gathers about the problem domain comes from the process worker either directly or indirectly, preferably the former.

The process workers[4] are the primary source of both information to solve the problem and feedback to confirm which of the various solutions will work best for them. The business analyst must have a good relationship with the process workers so that the information and feedback are forthcoming, useful, and free of bias.

The issue with process workers is that there are so many of them. The greater the number of process workers the larger the variety in experience, activities and tasks, longevity with the organization, ways of performing an individual activity, computer sophistication, preferences, prejudices, predilections, and so forth. Ideally, you want to investigate all variances that exist among the process workers. The more information you get from the process workers, the better able you are to:

- Identify all the ways the business process is being used today.
- Identify the most efficient method of solving the problem, which is usually some conglomerate of the various approaches.
- Get different perspectives on the problem (including opinions of those against the change).
- Incorporate as many of the variances into the solution as possible so that no process worker is disenfranchised.
- Obtain information about the diversity in the business community for transitioning to the new process.

Before you start interviewing or meeting with the process workers, there are some things to consider that apply to process workers, as opposed to management or other product stakeholders. Table 7.2 shows a comparison of the typical characteristics of process workers. During investigation, you want to analyze the information you receive based on the biases that process workers may exhibit.

TABLE 7.2 Process Worker Biases

Aspects	Bias
More computer sophistication.	May have an abundance of solutions for any given problem, therefore may be more intolerant of the process needed to define the solution.
Skew younger in age group than management.	May not be wedded to the old way of doing things.
	May not take into account the culture of the organization.
May be 9-5'er.	May give superficial answers and have poor participation.
	May not want to put in extra time in the solution definition effort.
May not be the most ideal candidate to discuss the process, but may be the only one available at the time.	May not have the right information but still answer questions.
Assigned to work with the business analyst at the sole discretion of the manager.	May represent only what the boss wants.

Sophisticated Users

When I started in data processing there were users so afraid of computers that they had to have their hands physically manipulated onto the keyboards of dumb terminals before they would believe that they could not destroy the computer, and that the computer would not attack them.

Nowadays users of computers and computer technology are sophisticated. Most have their own computers at home. Many know how to program these computers. A large number of the workforce today never worked in a business world that was not run by computers. They also do not remember a time before microwave ovens and color television, but that's another story. Because of their sophistication they expect more out of the computers they work with and are better able to imagine new features and functions the computer can do for them. They are not, however, necessarily better at describing these features and functions than were the previous generations of computer users, nor do they necessarily have a better grasp of the underlying technology.

When talking to the computer users of today, business analysts from IT have to be careful about condescension. For example, the following statement was overheard by a business analyst in a meeting with a user: "Yes, you may be able to do that at home on your PC, but we have much more complex technology here. Too complex to go into with you." Assume that any user you talk to may have wired her own home network and taken programming classes in school so you will approach each user with the appropriate respect.

Also, be wary of dismissing users' suggestions of solutions even when technologically infeasible. Behind each wild idea may be a problem-solving idea that the user could only relay through a proposed fantastic solution.

Non-Users

You are dealing with the entire problem domain. There will be non-IT activities in the business process. There are physical activities like moving paper from one area to another; there are authorizing activities in which paper must be signed; there are decision activities away from the computer; and so forth. This means that to understand the complete problem and all the impacts on the neighboring constituencies you must expand your purview from that of only the users of the computer system in question to all the process workers involved with the business process of which the computer system is a part. Include every process worker no matter how incidental the task they are performing seems to be. Try not to focus only on the computer system at hand. Remember the neighboring constituencies: those who provide the information to the target business process and those who receive information from it.

There are also process workers who are users, but of a different system than the one you are working on. They use the same data—perhaps data that is created in the target business process. They are computer users, just not users of the target system.

Enlarging your scope to include non-users increases the quality of your effort in several ways. For one thing, there may be activities or tasks that are necessary to the overall process, but are not known to those who only see the process through the automated system they are using. Process workers may identify manual tasks that could be automated, thus increasing the value of the overall product. Also you may discover ways of turning a process worker into a user by changing information flow to a neighboring constituency (see Chapter 15), thereby making the workers' jobs easier. And, it may enable you to solve one of the harder aspects of defining the complete solution: identifying the hidden and indirect stakeholders.

Classifying Process Workers

Typically, there will be a large number of process workers in the problem domain; too many to interview each and every one of them or even meet with all of them in groups. You will still want to get representative information from all of them to make sure you are defining the best solution for all. One way of doing this is called equivalence classification.

The concept of equivalence classes comes from software testing. The definition of an equivalence class is a set of data in which all elements of the set will act the same way, therefore one only needs to test one element. For example, if the minimum purchase for credit cards is $10 and the maximum purchase is $2,000, the tester creates a class of valid numbers between 10 and 2,000. The tester then can select $505 as a test, and does not have to test any other possibilities.

The concept applied to process workers is the same: Create classes of process workers such that the answers from any one member of that class will be essentially the same as any other member of that class. In doing so, you only need to conduct one information gathering session per class. This reduces the number of process workers you have to spend time with without reducing the quality of the information gathered.

The process workers can be classified in a number of ways to reduce the number of interviews without loss of valuable information. Some classification parameters include:

- Type and use of the current system or process:
    - Dedicated: needs the system to do the job. When the system fails, they cannot work.

TABLE 7.3 Process Worker Classification Matrix

	Shift	Longevity	Usage
Class 1	Day	Under 1 year	Dedicated
Class 2	Evening	1 to 10 years	Casual
Class 3	Night	Over 10 years	Discretionary

- Discretionary: uses the system when they need to. They are not dependent on the system.
- Casual: uses the system occasionally.
- Length of time using it or time in their position.
- Frequency of use.
- Different working situations, such as work shifts or geographic disparity.

Table 7.3 is an example of a process worker classification matrix.

The matrix shown in Table 7.3 defines 27 different interview classifications. That means a maximum of 54 interviews (assuming you did a one-on-one interview with two process workers from each category, one for information and a second for confirmation). That sounds like a lot, but compared to conducting 500 to cover the entire process worker population, it is not bad. To reduce the number of information gathering sessions, enlarge the size of classifications.

Managing Expectations

"How do I negotiate with the business to change their expectations? Or if you can't change them, how do you keep them in line with reality?"

An integral part of a successful relationship with product stakeholders and management is the management of expectations. There are situations where an apparently successful implementation was rejected because the solution did not meet some manager's expectations.

When major changes take place in the organization, politics and emotions may supersede facts. Expectations rule. Perceptions are truth. And projects may live or die based on expectations, perceptions, or both rather than results. Stakeholders' opinions about the success of the project may be governed more by whether the project met their expectations than whether the project actually solved the problem.

As a business analyst, those around you generally assume you are the manager of the expectations of nearly everyone involved. Many expectations are contained in unstated requirements or assumptions made during investigation. These expectations resurface during testing or in production. The more information that is transferred or documented, the more the expectations are on the table, the easier they can be managed. Here are some guidelines to follow to make it easier:

- To deliver 99 percent of what you promised only leads people to miss the remaining one percent, so be careful what you promise.
- Watch the campaign promises—the promises made during the initiation of a project or when the project needs to be approved that can set the wrong expectations.
- Be aware that if you meet high expectations through working late nights and weekends or other heroic efforts, the recipients expect more of the same. Keep all expectations realistic.
- Keep the problem statement in front of everyone as a simple statement of what to expect when the product is delivered.
- Capture expectations from meetings, conversations, and feedback and immediately address any new or changed expectations.
- Determine measurements for all expectations so you can prove the expectations have been met (i.e., acceptance criteria).
- When facing complaints or disappointment from the business community, focus on expectations rather than on the actual delivery. Expectations are like perceptions and do not always match reality.

Expectations are generally managed through communication and courage: communicating exactly what is going on and the courage to disappoint when the expectations outstrip reality. Managing expectations does not mean manipulating the truth or spinning the story.

You cannot manage expectations if you do not know what they are. The problem owner and others involved with the initiation of the project have expectations. The trick is to get the expectations out in the open so that you can manage them. This is done during Checkpoint Alpha discussed in Chapter 8.

When You Catch an Expectation, Confirm It or Correct It

One of the important concepts in managing expectations is to catch an expectation when it develops, or when you first identify it. Even idle speculation can turn into expectations and then demands, even when not included in a contract. It's easy to dismiss or ignore comments of expectations, especially when they conflict with stated agreements. You may believe that when

push comes to shove, the agreement, contract, or whatever will take precedence and the expectations will evaporate. Not always so. Even patently unreasonable expectations can gain a semblance of realism when allowed to fester unfettered. Any change to the solution that is suggested or even fantasized about should be questioned, tactfully of course, to keep the expectations within boundaries.

Saying No Gracefully

Another issue is saying no. It is not easy to say no to the business when they are requesting additional functionality in the system, especially when it means that they can use the system better or need the change to comply with some new regulation. After all, we are a service organization to the business. However, the business is not always right, and may not always have a good understanding of the scope and complexity of what they are asking. And, like children at Christmas time filling out their lists, the stakeholders may ask for everything they can think of to be sure they will get something. Since we do not want to have to say no to the customer, we might follow the advice the VP of management information systems (MIS) of a major telecommunications company gave me, ''It's never a matter of yes or no when they ask for changes, it's always a matter of now or later.''

One of the more efficient ways of saying no to a customer, problem owner, stakeholder, or process worker is to offer alternatives such that their request becomes a less attractive alternative among two or more choices, and let them make the choice. When you come back to the product stakeholder with a choice such as "you can have that requested change by adding an extra month on the deadline or you can get the problem solved on deadline and we can add the change later," the product stakeholder can weigh the benefits and consequences and make a reasoned decision. That way you don't say no, and still keep the solution intact. Also, as a tip, never respond immediately to any request no matter how ludicrous. Always "take in under consideration" or ''check it out.'' That way the product stakeholder feels as though you are valuing the request and when you come back with a response that is more negative than positive they will feel as though you have researched it and tried to make it happen and accept a negative response more readily.

Here are two examples of situations faced by business analysts.

Tummy Tuck Syndrome   Here is an analogy for the typical scenario that befalls the business analyst in dealing with the customer. Consider the business analyst as a doctor and the customer as the patient in the following:

The patient comes to the doctor with peritonitis. An appendec-
tomy must be performed immediately to save the patient's life.
The doctor, having a good bedside manner, describes the oper-
ation to the patient. There will be surgery; the doctor will open
up the patient's stomach, remove the appendix, and sew the pa-
tient back up with invisible thread that doesn't leave a scar, and
the patient will be good as new after recovery. The patient con-
siders this and then asks the doctor if, while he has the stomach
open, the doctor can do a tummy tuck. The doctor replies, "You
have two choices. I can take your appendix out and save your
life and in a few months when you've fully recovered we can
get together and discuss a tummy tuck, or I can give you a
tummy tuck now, and you'll have a really good looking corpse
at your funeral. What would you like to do?"

When the business community asks for the kitchen sink while you are
making changes or defining a new system, evaluate the request against the
problem. Does the request help solve it or not? When the request does not
contribute to the solution, the business analyst takes the same approach as
the doctor. The first question is, "Does this go to solving the problem?" If
not, then the discussion is over. The feature must wait for the next round or
next release. If the requestor can show that the feature does go to solving
the problem, or insists it be included anyway, the response is, "Here are
your choices: We can get the problem solved in the timeframe with high-
quality results or we can add the feature and either come behind schedule
or with a lower-quality product. Which would you prefer?"

Gilding the Lily[5]   On the other hand, the solution team often goes off on its
own under the banner of "we know what's best for the users." Sometimes
the innovations the developers come up with do produce better results for
the users and the whistles and bells they create become part of the solution.
These unrequested enhancements or modifications are wonderful if they
can be done within the constraints of the project's time and cost. Even so,
the business analyst still has to update the requirements and take the
updated version back to the authorizing person for re-approval. Remember,
there should be no surprises, pleasant or otherwise at time of delivery. Many
times, the suggested change is simply not applicable because the developers
have not spent the time to understand what the users really want and really
do in their environment. The business analyst has to be just as assiduous
about keeping additional features from being added to the product by the
development side.

One technique to make sure no extra gold plating takes place that may
have an effect on the product stakeholders at delivery or acceptance, is the

Checkpoint Charley meeting discussed in Chapter 15. In this meeting the solution team presents the implementation solution to the business analyst so that the business analyst can make sure changes are presented early to the product stakeholders. Also the business analyst may catch misunderstandings in the solution implementation before the design turns into code.

The theme is collaboration. You play various roles in the organization and work with the various players involved for a single goal: to solve business problems. Each of the roles you play and each of the players you interact with have a place and purpose in the overall process of defining and implementing a solution that increases the value of the organization you are working for. The more the collaboration, the better the solution. Defining who you are and where you are in the organization is the first step, and identifying those you will be working with and how to work with them is next. Now let's look at how we are going to solve those problems. What is the process to follow to increase the changes of successful problem solving in the business? Read on.

Notes

1. Project Management Institute, *The Guide to the Project Management Body of Knowledge, 4th Edition* (King of Prussia, PA: Project Management Institute, 2008).

2. Scott Ambler, "Scaling On Site Customer," *Dr. Dobbs Journal* 63–66 (January 2008).

3. International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge,* version 2.0 (March 31, 2011), 11.

4. We are expanding our view of the user to process worker. In this way, we are looking at the business process rather than just computer functions. The user is a subset of process worker and defines those who specifically work with a computer system. Any process worker who is not a user could legitimately become a user with the changes the business analyst makes.

5. When the users or other stakeholders add some features or modify some requirements, it's called scope creep, which is something ugly. However, when the designers or programmers add in a little something extra, it's called gilding, which implies making something more attractive. You can guess who came up with these labels.

The Problem

The formulation of a problem is the most essential part of problem solving.

—Albert Einstein

It is simple: You have to know what the problem is before you can solve it. And you need to know that the people for whom you are solving the problem agree with your concept of the problem.

Is it really the job of the business analyst to diagnose the problem? Can't the businessperson tell us what the problem is and we just have to focus on defining what must be done to solve it? There are two issues. First is the assumption that the businessperson knows what the problem is and can, or is willing to, define it for us. We discuss that in Chapter 8. The second issue is how we see ourselves in the overall relationship. Let me use an analogy, since we are discussing diagnosing the problem.

The Business Internist

According to the Random House Dictionary, an internist is defined as "the medical specialty concerned with the diagnosis and nonsurgical treatment of unusual or serious diseases." The internist is not the doctor who operates. The internist diagnoses the problem and recommends the treatment that is carried out by the specialists. Prior to prescribing a course of treatment, the internist gains knowledge of the problem domain (the body) and the symptoms of the problem (the pain and physical complaints). Then the

internist makes sure that the patient knows exactly what is wrong and what treatments (solutions) are available. The internist makes sure that the patient understands that there is a problem, what that problem is, and the severity of that problem. Unless the patient understands everything about the problem, and the options for solution, there will be no action taken. Once a treatment or solution is selected, the internist turns the case over to the appropriate specialist(s). The internist is also concerned with the management of the cure. From this perspective, the internist checks on the progress of the treatment to make sure that the assigned doctors and support staff are applying the correct treatment and that the patient is healing appropriately. While the specialists are out of the picture once the operation is complete and successful, the internist still consults with the patient after the cure has taken place to ensure that the patient has a full recovery from the problem, the solution is permanent, and there are no side effects or reversals.

The internist also conducts a number of tests to determine the cause of the illness or problem. The metrics that result from these tests serve to identify the real problem and potential solutions, and as physical evidence to the patient of the severity of the situation. A patient may not want to change his diet to lower his cholesterol until he has seen the results of the blood test, even if the patient has absolute confidence in the internist.

Internists are also consultants. From the American College of Physicians: "Internists are sometimes referred to as the 'doctor's doctor,' because they are often called upon to act as consultants to other physicians to help solve puzzling diagnostic problems." In the ideal organizational setting, business analysts are considered internal consultants, assisting the business to solve problems with the application of their analytical skills and expertise.

Internists are trained to treat patients as whole people, not on a disease-by-disease or symptom-by-symptom basis. The same holds true for the business analyst. The business analyst does not look at any one department or issue in the organization independently as a single problem. The business analyst views reported issues from the perspective of the entire organization, examining impacts to other parts of the business, the value of solving the problem, and whether there really is a problem or if the issue is a form of business hypochondria. This holistic, totally objective, and independent approach to solving business problems is the hallmark of the business analyst.

Internists are trained to recognize situations where several different illnesses may strike at the same time. Similarly, the business analyst does not restrict herself to just the issue presented by the business, but examines all the symptoms and defines all the problems, those interrelated with the issue and those unrelated. In this way the business analyst can present to upper-level management the total picture for their decision making and also uncover additional problems that need solving.

Gathering information about the problem, analyzing that information, diagnosing and stating the solution, and identifying the correct specialist to supply the cure: that is the essence of the business analyst.

The business analyst acts the same as an internist solving business problems rather than medical illnesses. The business analyst takes measurements of the current operations, examines the situation, asks questions, advises the problem owner on what the real problem is, sends instructions to the specialists in IT in the form of requirements, answers questions during the curative procedure, and checks with the business after the problem is solved. Of course, the business analyst may have a number of people who are the patient suffering from the problem, where the internist deals with one at a time.
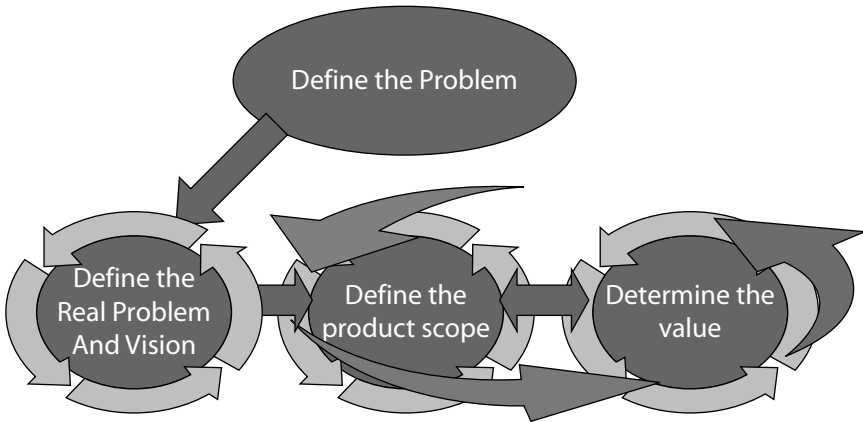
The Process of Defining the Real Business Problem

Defining the problem is a process, part of the overall systems approach, in which we define the problem and all its components, and determine whether or not the problem should be solved. There are three elements to this process as shown in the graphic. Each of the elements is iterative in itself. Once the problem and vision have been defined, the remaining two elements impact each other as shown. These three elements are:

1. Define the real problem: Gather information about the problem domain and the current conditions, determine what the problem might be, analyze that information to define the real problem to be solved and determine the vision of the solution. We discuss this process in Chapter 8.
2. Define the product scope: The product scope delineates the problem and the solution. The product scope defines what the solution team, or the project, will produce, and provides the essential decision making

information to the business case and/or project charter. The product scope is discussed in Chapter 9.

3. Once the problem has been defined we need to determine whether it is worth spending the time and money to solve it. The solution must be in alignment with the organizational mission, goals, strategies, policies, and it must have a value to the organization in excess of the cost of solution. We discuss this element in Chapter 10.

Define the Problem

Here is a paragraph from the introduction to the initiation phase of a business process written for a financial organization in New York:

> Each project, a single change request or combination of change requests, must have a business problem that it solves. The initiation phase determines the business problem and all the associated information necessary to produce the business case and project charter so a decision can be made to initiate a project.

As shown in Figure 8.1, the first step in the business analyst solution cycle is to define the impetus for the effort (the problem) and a depiction or scenario of what we wish to achieve with the effort (the vision). These two elements are the basis for the product scope and the entire solution effort. They define where we are and where we want to be. This chapter addresses that activity.

First Things First

"We don't get good requirements because we do not know the problem."

When the business community describes the problem to solve they typically describe several problems, all of which they would like you to solve with a single project. Your responsibility, as business analyst, is to solve one problem, so you have to determine which of the described problems is the real problem. Many times the reported problem is vague, ambiguous, uncertain, or misleading. This is not a negative comment on the ability of users of computer systems to report issues accurately. This is common when reporting problems of any kind under any circumstances in or outside of
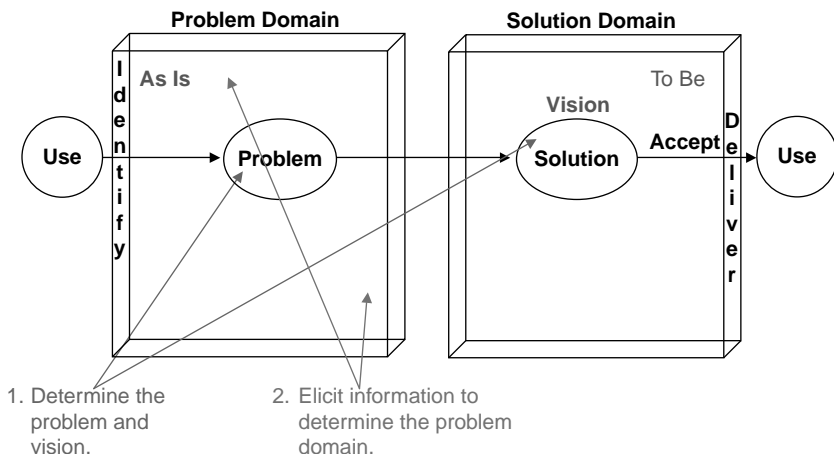
FIGURE 8.1 Step 1: Determine Problem and Vision

business. Consider the earlier example of the internist. If you were able to describe your medical problem succinctly, accurately, and completely, you would not need a doctor to intervene to diagnose your illness. You could simply call on the specialist to provide the cure.

The business analyst's first task is to define the real problem that the organization needs solved. This sometimes means completely ignoring the stated problem and conducting an investigation. The business analyst plays the role of investigator to do this: observing the business area, asking questions, and analyzing the information. Once the real problem has been defined, it is a good idea to check with management to make sure that the defined problem is still one which management wants to solve at this time. There are times when management's view or expectation of the problem is not in line with what the real problem is, and the real problem is too big to solve or simply not worth solving.

Problems

"We do not get good requirements because in the time it takes to get the project approved, the problem changes."

🌐 Case Study

At a large New York financial company, management set a goal of tripling revenues over an eight-year period. Everyone charged ahead on the 3X project. For more than two years changes were made,

software written, ideas were tried, data and processes reorganized, and yet upper management became concerned about the apparent lack of progress in this highly visible project. At a meeting to get a handle on the number of changes and adjustments and reversals going on, a business analyst asked a simple question: "What are the problems that prevent us from tripling our revenues?" As the team discussed the question and a wide range of responses were recorded, they realized that while they were making some improvements, they were tackling symptoms and applying Band-Aids and implementing what amounted to a temporary fix. The team stopped all work for a few weeks and continued to meet to define the real problem that needed solving. When the team stopped running projects and focused on defining the problem, they came up with what really had to be done, and within six months showed more positive results than in the past two years combined.

What Is a Problem?

The Merriam-Webster online dictionary defines a problem as a "question raised for inquiry, consideration or solution." For our purposes we define a business problem a bit more specifically. A voiced problem may start as a question. The real problem behind the question that arises from the investigation and analysis by the business analyst is best understood by this definition:

> *A problem is the difference between someone's desires and the way things seem to be.*
>
> —Gerald Weinberg, *Are Your Lights On?*

Gerald Weinberg's definition of a problem is certainly elegant and concise. By using the words desire and seem to be, he introduces the concept of perception. A perception means that there may not be a real problem in play and that the perceived solution (desire) may not be the best or even correct solution. By dealing with perceptions rather than desires, defining and solving the problem are less of an emotional and personal issue. We can disabuse someone of a perception; it is harder to change someone's conviction.

Many organizations divide the business analyst's work into addressing problems and opportunities. A problem indicates that something is not right. An opportunity means that things are operating well and could be improved. Gerald Weinberg's definition of problem covers both defects in the operation and desired improvements under the same banner of problem. In both cases someone's desires do not match with perceived reality.

When that pesky blue-bordered dialog box pops up on the Web page telling you, "Warning: Unresponsive script," it is telling you there is a problem which needs to be fixed. When the VP of marketing wants the product sales report displayed on the screen instead of being printed on paper, they are also identifying a problem. (Note that the VP is specifying an enhancement or modification to an operational system, and that the problem might be that printed reports are not environmentally friendly, or that the printed reports are not as current or accessible as a screen display, or that they keep misplacing the hard copy report.)

Challenge 1: Finding the Problem

The following is a short list of reasons why we cannot determine the real business problem by simply asking the stakeholders. Business analysts stated these reasons over the past several years:

Noise, static (other problems interfering with the real problem or politics).

Improper training—system or process already has solution, but user does not know it is there or how to use it.

Conflicting solutions—solution may cause another problem or be perceived to do so.

Does not appear to be a problem to the one you are talking to.

User only knows his perspective.

They have other vested interests.

Only see the symptoms.

May just need something better so does not raise the issue as a problem.

The business presents solution and not problem.

Know the problem, but cannot describe it accurately.

Don't really know what the problem is.

Sometimes business managers conceive problems and require solutions to overcome their inability to manage.

Being afraid to admit what the problem is might indicate incompetence or politics.

Solution seems obvious or already exists.

User cannot express what the problem is—"It just isn't right."

Currently using a work-around so there is no perceived problem.

Is not directly involved so does not see the problem.

Only seeing one facet.

Not aware of the real problem.

May not really know.

*Does not know the problem; it just does not feel right.

Might be presenting the problem in wrong terms.

Too close to the problem to see it.

They may be the cause of the problem.

The problem is a conglomeration of several things so it cannot be defined as a single problem to be solved.

Afraid of the solution—"band-aid the break."

*User cannot express what the problem is—"it just isn't right."	Too many alligators—too much emotion, too much detail, too many factors.
May be looking at surface issues only.	Doesn't care about the problem or solving it.
Afraid of change.	Solution may not be beneficial.
It is opinion not fact.	Does not know all the information.

Then there are the political reasons for not exposing the real problem (also from business analysts):

- Hide the problem because it could reflect badly on them.
- Deflect the problem to reduce its seriousness or blame.
- Point the finger at or blame another person or department.
- Afraid of getting blamed for the problem whether it is their problem or not.
- Try to hide the problem to avoid or shift responsibility.
- Fear of retribution.

Finally, some product stakeholders simply have this attitude: "I am not sure what my problem is, but maybe if I complain long enough and hard enough about the issue, someone else will come up with a solution and rectify the situation." That someone else is the business analyst. Initially, the product stakeholders have great expectations of the business analyst. The BA represents the end to the problem and associated symptoms, even if no one really knows what the problem is.

Regardless of the type of problem and how well it may be intentionally or unintentionally hidden, it is the business analyst's job to ultimately locate the real problem.

Challenge 2: The Unstated Problem

The first assumption that a business analyst typically makes is that the problem presented to him is the problem that must be solved. After all, it is the business that identified the problem, and the business should know what the problem is. Moreover, the business is the customer, and the customer is always right.

You might not hear from those with the real problem. At an accident scene those with the real injuries are the ones who are *not* complaining. The EMTs are tempted to treat those who are complaining because the complainers can tell them where it hurts and they can provide the quick fix.

In reality, the ones who are unconscious are probably the ones in greater danger. Similarly, the real business problems that you need to address may be hidden and not spoken about.

Many times those with the problem simply don't say anything about it. They may assume that it is someone else's problem to solve, or they are simply afraid to bring it to the attention of management. Even when you ask, they may be reticent to admit there is a problem for fear of being a whistleblower or tattletale, or, even worse, a rat.

Assuming upper-level management has a handle on the problems is a risky assumption. We previously discussed that upper-level management is focused on strategic aspects so they may not have any clue about tactical problems until lower-level management brings it to their attention. And by then the description of the issue may be so watered down that the real problem is lost.

The problem is often stated at a high level of granularity so that any solution will address some of the problem, and no solution will completely solve it. For example, the statement, "The problem is that the accounts payable system is not efficient" might be a problem to someone, and they may have the statistics to prove it, but the real problem to define is why the accounts payable system is not efficient.[1]

The most insidious aspect of problems is the work-around. We have systems with work-arounds that were put into effect 10 years ago and have become so engrained as part of the operations that no one can even remember the problem the work-around solves. Temporary patches have been in place for 20 years because no one has time to make the permanent fix and besides, the problem is solved anyway. This is the IT version of the rain/roof paradox: When you have a hole in the roof, there is no need to fix it when it is not raining because it isn't a problem, and when it is raining you can't go up and fix it because it is raining.

Even worse is when the problem is accepted for better or worse. Process workers get tired of complaining about something that apparently management doesn't see fit to fix. Or, they assume that the problem must be unfixable, or at least unfixable at a price the organization can afford. Or they figure it's something that is inherent in computer technology or in the specific system, something they have to live with, and they do. And the problem is not spoken of again.

Challenge 3: The Misunderstood Problem

Of course usually we have a stated problem to deal with. The issue is whether it is the real problem we should be solving or not. Certainly management wants enthusiastic, energized team players jumping on every

project with a will to win. However, when the project is ill-conceived or simply not worth doing, all that energy and enthusiasm is wasted or aimed in the wrong direction. So instead of charging ahead on every order, the business analyst should be questioning every project to determine why this project is being run.

Here is a fanciful example:

Alice, our senior accounting person, has reported up the chain of command that she is having a problem with the blue screen of death on the new accounting system. Don, the Director of Finance, who reports to the CFO, asks you to talk to Alice to solve her problem. You dutifully go to her office to fix her computer or provide some ad hoc training that will prevent future blue screens of death from appearing. When you get there, you notice that the entire office is decorated in blue. You ask the usual questions to determine the cause of the problem. She explains that it is not the screen of death that bothers her. She just reboots and goes on about her life taking time out for a cup of coffee. What she is having a problem with is the *color* of the screen of death. She hates the blue. What she wants is to be able to change the color of the screen of death so that when it appears it will match her outfit: yellow, green, pink, chartreuse, and so forth.

Don interpreted Alice's stated problem as potentially representing a generic failure in the accounting system, or perhaps a failure in training. Once you inform him of the real problem—the color of the screen—he decides the problem is not worth solving.

When the business analyst acts upon the problem statement received from the business without investigation to confirm the real problem at hand, the solution team may end up providing an elegant and complete solution to the wrong problem. The business then blames IT for not solving the problem. IT then blames the business community for not knowing what they want.

Define the Real Problem

"We didn't get good requirements because the problem was not completely defined."

So, okay. We won't necessarily get the real problem from the business community. And we can't depend on the stakeholders to define any problem, much less the real problem. Then, what do we do? How can we

determine the real problem from the complaints, symptoms, and disgruntled chatter?

This is from an insurance company business analyst process:

3.4.4.1.2 Problem Statement

The problem statement states what the problem is. The statement should be clear to anyone reading it without needing additional technical explanation. The problem is usually stated in negative or limiting (constraining) language.

This is from a process written for a banking organization:

The business analyst elicits information from the requester [of the change] to determine what the business problem is that needs solving and why the business is interested in solving the problem. The business analyst gathers information to determine what change is requested and to determine the initial impacts the change will have on the software, data, operations and business.

Ask Questions

The focus of the first information gathering process is on the problem owner and business area management. Questions you ask are context free—that is, they are questions that can be asked in any situation and are not specifically related to the problem at hand. By asking context-free questions, you are reducing the chance that the responses will be in the form of solutions rather than problems.

During the problem-definition process, here are the questions you want to get answered:

- What is the real problem we want to solve?
- What is the business justification for solving the problem?
- What are the risks associated with the issues?
- What if we do not solve the problem, or do not solve it within the deadline (if a deadline is stated)?
- What are the impacts to the business for any given solution?
- Are there any constraints from the business affecting the way the problem is solved?
- Who is affected by the problem?
- Who owns the problem?
- When does the problem occur (intermittent, constant, chronic, etc.)?
- How long has the problem been going on?

- What does it look or feel like when the problem is solved (what is the vision)?
- How will we know that the problem is solved?
- Where in the organization does the problem exist?
- How do you know it is a problem?
- When did you realize it was a problem?
- What is the alignment of the problem? What business strategy, objectives, and so on is the problem or opportunity related to?
- How does a solution fit with organization strategies?
- Who are the decision makers who influence the solution?
- Why is it necessary to solve this problem?

The Five Whys Technique

One of the techniques you can use to uncover the real problem is the Five Whys. The technique was developed by Sakichi Toyoda and used at the Toyota Motor Corporation. It was later adopted in Six Sigma practices. The technique is used to distinguish symptoms from root causes by asking "Why?" five times to get to the root cause of any problem. It is basically a reminder not to let the first answer suffice as the reason for something.

For our purposes, the technique is used to generate more information even if it does not get to the root cause. One of the drawbacks of the technique is the knowledge of the responder. If the responder does not know the root cause—it is not within the responder's awareness—all the whys in the world will not unearth the root cause. Here is a typical scenario:

Charley: We have a big problem with accounts payable.

Business analyst: What is the problem?

Charley: The voucher entry system does not work anymore.

Business analyst: What do you mean by 'does not work'? (Why 1.)

Charley: It takes too long to enter all the vouchers.

Business analyst: Why does it take too long to enter all the vouchers? (Why 2.)

Charley: Because there are still vouchers left at the end of the day.

Business analyst: And why do you think that is a problem? (Why 3.)

Charley: Because I have to work overtime to get them all done before I leave for the day.

Business analyst: And why is that a problem? (Why 4.)

Charley: Because I don't get to happy hour on time.

Business analyst: Why don't you get to happy hour on time? (Why 5.)

Charley: (exasperated): Because it takes too long to enter all the vouchers.

And the questioning starts to sound like an Abbott and Costello routine. Charley simply does not know why he is not getting to happy hour on time. He just knows he has a problem.

There are a few issues with this technique to be aware of:

- When you happen to be talking to a person who has a two-year old, the successive why questions could expose a sensitivity that might not be appreciated.
- Before you get to the problem or the fifth why, the responder might stop the questioning with a simple, "I don't know." (I don't recommend asking "Why don't you know?")
- The question "Why?" can be interpreted as a challenge, as in, "Why on earth would anyone do *that*?"

When using this approach, ask the why question in different ways. To reduce the implication of challenge in the why question, try phrasing it this way: "Why do you think . . . " That not only removes the challenge, but makes it clear that you want the person's opinion and forestalls an answer like "You'd have to ask someone else that."

Another question to ask to increase the flow of information without risk is "How do you know?" In the example, Charley interpreted the "Why does it take too long to enter all the vouchers" as "How do you know it takes too long?"

Another way of discovering, or at least getting closer to the real problem is to reverse the question. Instead of focusing on what the problem is, as what the solution might look like: "What would it be like if you [didn't have this problem]?" Sometimes the problem becomes clear when the solution is stated, especially in terms of a scenario—for example, defining the product vision, which is discussed later in this chapter.

Analyzing to Identify the Real Problem

We cannot line the potential problems up in a row, stand behind the one-way mirror and have the witness point to the real problem we have to solve. We have to bring our analytical prowess to bear to determine what the real

problem to solve is. One way to do this is solicit help with a process I like to call the problem determination game.

The Problem Determination Game

This game is played with other business analysts whether or not they are familiar with the problem area. There are six steps in the game. In the first step, we gather a list of potential problems. The remaining steps reduce the number of potential problems by analyzing the problem candidates against various criteria. The object is to arrive at only one problem statement at the end of the game. That statement is the real problem. Here is how the game is played.

Step 1: Gather All Possible Problem Candidates

Not everyone will have the same take on what the problem is. You will get a different spin, a different viewpoint, a description of a different problem altogether from each person. The business community will make many statements purporting to define the problem. The problem owner and other stakeholders may also introduce several problems to solve with this one project (i.e., tummy tuck syndrome). Each of these is a problem candidate and may be the real problem. Some problem statements conflict with others, which reflect the different perspectives those involved with the problem have. You can only solve one problem at a time, so you need to determine the real problem you are going to solve.

There is another dynamic at play when you start looking for the real problem. Once invited to express problems, people will tell you *all* the problems they are experiencing, regardless if the problems are pertinent to the issue at hand. This is not necessarily bad. First of all, the more problem candidates you elicit the more likely you are to get the real problem. Second, new problems may surface that need to be addressed and you can bring them to management's attention, as long as the additional problems are not lumped in with the one you are going to solve.

Each voiced problem has a meaning even though only one can be the real problem you will solve now. The others may be requirements, constraints, statements of risk, or problems for someone else to solve. They may reveal hidden agendas on someone's part. In all cases, a statement of a problem says that the issue is important to the person stating it.

In the case study, "Determine the Problem Candidates," note the different views of the same problem by different levels of the organization. These problem statements are the result of elicitation sessions held with each of the responders. As a business analyst, you are not going to automatically assume

that the CFO's response is the correct one just because of his position. Your job is to determine the real problem.

Now review the problem statements offered by various managers. Which of them do you think is the real problem we have to solve? Did you pick one? Good. Let's continue the game and see if you are correct.

Case Study

Determine the Problem Candidates

Mary, the A/P manager and problem owner, has requested a business analyst to solve her problem with the A/P system. You dutifully elicit information from several sources who give you the following responses:

Mary (A/P manager):

The problem is that there is too much voucher entry rework.

The vouchers are not entered correctly.

Payment calculations are not being performed accurately.

It takes too long to set up a new vendor.

We are getting too many complaints from the vendors that we are not paying their full invoices on time.

Susan (supervisor):

If I do not get the overtime under control we will bust the budget.

The voucher entry process is not fast enough.

The turnover in my section is too high.

The monitors are always breaking down.

Charles O'Brian (CFO):

We have too many vendors.

The A/P system is not efficient.

We need PCs instead of the legacy terminals we have now.

We are overpaying the vendors.

> Donna Anderson (Director of Accounting):
>
> The current system does not have the ability to handle new vendors' terms.
>
> We are not going to be able to attract or keep our vendors.

At this point assemble the business analysts for the game. You might warm them up by brainstorming what they think the problem might be and adding each of their suggestions to the list.

Step 2: Determine Problems That Are Out of Scope

Make sure that the problem is something you can actually solve. During analysis you ask: Can anyone involved with the project do anything about this problem?

If answer is "No," then it is not your problem; cross it off the list.

With this step you are trying to eliminate problems that the business might want to foist on IT for a solution that cannot be solved by IT or by changing a business process. You can detect problems in this category and eliminate them. Here are some examples of problems that should be eliminated from the list:

- "The suppliers are charging too much for maintenance of our equipment, so we need to change suppliers."
- "We need a better compensation system for our salesmen."

Case Study

Problems That Are Out of Scope

The CFO noted, "We have too many vendors."

The number of vendors is not something IT can do anything about. That is a corporate policy or strategy. IT only records the vendors the corporation has. This problem is removed from our list.

The issue of turnover in the voucher entry department might also be out of scope. It might be due to an old boring system approach and improving the user interface might reduce turnover by making the job a bit more interesting. Since there is a doubt, you leave the problem on the list, for now.

Step 3: Determine Who Owns the Problem

Make sure the problem you are solving is one for which the target business unit will pay. The analysis questions you ask are:

- Is the owner someone within the organization?
- Is the owner the one asking for the solution?

If the answer is "No," then it is not your problem; cross it off the list.

At this step, you are concerned with the authorization for the problem. For example, when the accounts receivable department is paying for the solution, you do not solve problems for the marketing department even when marketing problems come up during your investigation. While you certainly care about the problems that marketing has, the focus now is the accounts receivable department.

Many times you will get a problem reported by a business unit that does not exist in that area. For example, data that is reported as being erroneous in one department is actually entered incorrectly in another department earlier in the business process. When you eliminate the problems that belong to some constituency other than the one you are assigned to help, remember to capture the problem statements and inform the appropriate department or constituency about the reported problem so they can determine whether to solve it.

Case Study

Problems the Target Business Unit Does Not Own

Both the CFO and Susan complained about the old equipment being used for the A/P system. Susan expressed a problem (the terminals break down too much) and the CFO stated a solution (solve the problem by installing PCs).

Since IT owns the computer equipment, the accounts payable department cannot solve the problem. These problems belong to IT.

Step 4: Determine Relevancy

Make sure that the problem exists today and not in the future. When the problem is not current, it is not a problem (yet). When the problem is temporary or anticipated (it will occur when some action is taken), then it is not relevant.

The question you ask during analysis is: Are you currently having this problem?

If the answer is "No," remove this problem candidate from the list.

We only want to deal with problems that exist *now*, or in the foreseeable future. A problem that may exist in the future based on conditions that may change or actions that may or may not be taken, is not a problem we can solve today. Usually a problem statement beginning with the word "if" falls into this category. For example: "If we are not compliant with that new regulation by September 1, the CEO will go to jail." The CEO going to jail is not a problem except perhaps for the CEO. The real problem is that the company is not compliant with the new regulation. Another clue that the problem is not relevant is when it is stated in future tense, as in "the problem is that we *will* lose sales if we don't . . . "

Capture the problems that fail the test of relevancy. They usually represent product or business risks.

Case Study

Problems That Are Not Relevant

Susan stated, "If we don't get the overtime under control, we will bust the budget." As of now the budget is not busted and you certainly intend to get it under control. "Busting the budget" is not a problem.

Donna mentions the potential loss of vendors. This is not happening now, otherwise she would have said it in present tense.

Both problems are eliminated as problems. They are kept in a list of risks.

Step 5: Distinguish between Problems and Requirements

By definition, problems have more than one solution. To eliminate any problem that really is not a problem, ask the following question: Can anyone identify a solution to this problem?

If there is only one solution, the statement is an implied requirement, or a constraint on the solution and you can eliminate it from consideration as the real problem.

Here is an example: A VP has a problem and says they need a monitor with a 50-inch screen to view all their reports at the same time. There is only one way to solve the problem of delivering a 50-inch screen. There are a dozen ways to solve the problem of viewing all the reports. The 50-inch screen is either a solution or a constraint, not a problem. The problem is the

VP cannot view all their reports at one time; however, we will need to know why it is necessary to view all their reports at one time. (We may find out that the need to display all her reports is simply a justification to acquire a 50-inch monitor because the VP down the hall has a 48-inch monitor.)

One clue to determining problems to be eliminated is the occurrence of adjectives or adverbs in the problem statement. When the problem statement says ''The process is not flexible,'' there is only one solution: Make it flexible. To know the real problem you need to know why it is not flexible.

When you discover problems that have only one solution and should be taken off your list, examine them. Each removed problem may be a suggested solution, a business or product constraint, a political success factor, or a requirement. You do not have to make any determination at this point—your goal in this game is to define the real problem.

Note in the case study, ''Removing Non-Problems,'' that in addition to removing all of the CFO's potential problems, you also removed all but one of Susan's issues, and the remaining one—turnover—is suspect. This is not unusual. The CFO is too far removed from the problem and Susan may be entirely too close.

🌐 Case Study

Removing Non-Problems

The CFO mentioned a problem with efficiency. There is only one way to solve that problem: Make the system more efficient. You allocate this comment to the requirements list and make a note to define later what he means by efficiency.

You also note that this removes all of the CFO's problem statements. He does not have a handle on this particular system.

You also eliminate Susan's issue with the speed of the voucher data entry that is also a requirement.

Step 6: Distinguish between Problems and Symptoms

Assess the remaining problem candidates to eliminate symptoms. Now, for each remaining problem statement we ask another question: When the problem identified in this statement is solved, do any other problem statements go away? Remove any problem candidates that are completely solved when any other problem is solved. These are symptoms.

This step removes most of the problems from the list. You hold this one until last for two reasons:

1. This step takes a bit longer than the other steps, so the fewer problem candidates you have left, the faster it will go.
2. The final result and the actions to be taken are based on this last step.

Case Study

Problems left at this point:

1. There is too much voucher entry rework.
2. The vouchers are not entered correctly.
3. Payment calculations are not being performed accurately.
4. It takes too long to set up a new vendor.
5. We are getting too many complaints from the vendors that we are not paying their full invoices on time.
6. The current system does not have the ability to handle new vendors' terms.
7. The turnover in my section is too high.

You know you can remove some expressed problems immediately. For example, loss of customers is not a problem regardless of how it is stated in the executive offices. Loss of customers is a symptom resulting from some other problem (noncompetitive pricing, lack of new products, poor customer service, etc.). The same holds true for loss of sales, loss of revenue, loss of profit, increase costs, and so on. Of course, you probably should not challenge the CEO who announces that the problem causing poor quarterly earnings is a drop in sales.

Note that of the remaining problem candidates, most of the survivors are from Mary, the problem owner. This is typical since the problem owner should be the person most familiar with the problem at a high enough level in the organization to address it.

From the list of remaining problems select one problem with which to start. Compare that problem to all the rest to see if any other problems are completely solved when you solve the problem you selected. When they are, cross them off the list. Then go on to a second problem candidate and

repeat the process with the remaining problems including the first one. Continue in this manner until all problems have been compared with all other remaining problems.

Case Study

Eliminate the Symptoms

Suppose you start with the last stated problem: turnover (#7). When we solve the turnover problem so that voucher entry people stay with the company for years (increase the pay, increase the benefits, etc.) will that solve any other problem listed? It is unlikely.

Then you select #6. When you add functionality that allows the flexibility to put vendors on the vendor database with all the variant terms of payment:

- The voucher entry rework (#1) as a result of not entering the right vendor terms would be reduced to none.
- The vouchers would be entered correctly (#2).
- The payment calculations (#3) would be correct when the payment terms are correct.
- Vendor set-up would be faster (#4) because the vendor entry would include all the terms so the data entry people do not have to research and work around exceptions.
- Vendors would be paid on time (#5) because the terms would be correct.

What about the turnover? You determined earlier that turnover is only your problem to solve when the turnover is due directly to a poorly designed computer system or user interface. You will most likely be redesigning the user interface to correspond with the changes made to the vendor master file so problem #8 might be solved; otherwise, it is not your problem.

When you complete this analysis, you will have one of the following three options:

1. You have only one problem candidate left—the most common experience. You now have the real problem and you can move on.

2. There are multiple problems left—the second most common result. Review the remaining problems to ensure that all are mutually exclusive. When the remaining problems are all mutually exclusive, these are your alternative approaches:
   - Bring problems to upper-level management to determine which problem to solve.
   - Combine the two problems into a single problem statement without the use of a conjunction (using a conjunction means that you are still dealing with two separate problems).
   - Get more information and repeat the game.
   - You can only legitimately solve one problem at a time. The other problems can be given to others to solve or placed in a queue to solve after you finish this one.
3. There are no problems left—the last two cancel each other out. When this happens, these are your alternative approaches:
   - Combine those that cancel each other out into one single problem statement that includes both.
   - Get more information and repeat the game.

In our miniature case study, the problem is in the current definition of the vendor database. The vendor database as currently constituted does not allow for enough variance in vendor terms of payment. The result is slow payments, extra work, and rework trying to make new vendors fit the current definition, and an increased number of mistakes and inaccurate payments. Oh, and also excessive overtime that causes Charley to miss happy hour.

Why Bother?

Suppose you did not bother to define the real problem and took the CFO's definition of the problem and worked with IT to replace all the 3270 terminals with PCs? You would have made the voucher entry a bit better, brought the A/P department into the twenty-first century, reduced the number of broken monitors, and perhaps lowered some support costs. However, you certainly would not have solved the real problem of paying the vendors correctly and on time.

Suppose you focused on the local problem: Susan's overtime issue and the cumbersome voucher entry process? You speed up the process on the 3270 terminals, and reduce overtime, getting Charley to happy hour on time. Temporarily. Eventually the real problem with the A/P system will come back and Charley will be complaining about being late for happy hour again.

Documenting the Problem

Take care with the final wording of the real problem once you have defined it. The problem remains the same for the length of the project until the solution is in place and the problem no longer exists, or until conditions change in the problem domain that render the problem no longer applicable.

Regardless of the format you use to define the problem, here are some guidelines:

- The problem should be stated in negative or constraining language (see the case study, "Language of the Problem," for example).
- The problem statement should be as unambiguous as possible, not implying any other problem or dependent on the solution of another problem.
- The problem should be stated in as clear and concise language as possible.
- Everyone involved—product stakeholders, solution team, upper-level management—should understand the same problem in the same way. The problem may mean different things to different constituencies; for example, to Susan it means she does not have to renegotiate for more overtime budget; to Mary it means reducing vendor complaints; to Charley it means getting to happy hour on time.

Case Study

Language of the Problem

For example, compare the two potential problem statements for our mini case study:

1. We need a new vendor database definition that includes better vendor terms.

Or

2. The vendor database as currently constituted does not allow for enough variance in vendor terms of payment. (The results of this are slow payments, extra work, and rework trying to make new vendors fit the current definition, and an increased number of mistakes and inaccurate payments.)

Which problem statement do you think would provide more motivation to the developers to solve? Which is easier to sell in the corporate boardroom? Which constrains the scope to something reasonable and realistic?

Bottom line: Do not act on the request for a solution until you have defined the real problem and gotten acquiescence from an organizational authority that the organization wants to solve this problem.

The last item in our problem definition process is the confirmation. Even when we verify that the problem as defined by the problem owner is in fact the real problem, we still need to get it confirmed by someone in authority, unless of course that someone in authority is also the problem owner. This is called Checkpoint Alpha, and is discussed later in this chapter. However, we do not really want to take a problem to management without a corresponding solution, and we do not have a solution at this point. What we can provide is a vision.

Product Vision

Vision without action is a daydream. Action without vision is a nightmare.

—Japanese Proverb

Once you have the problem you have a starting point. You know what has to be done, assuming, of course, the problem is one that needs solving. Now you have to determine where you are going, what the end of the road looks like. That is the vision. There is always a vision attached to any problem. When you realize you have a problem, you also know what it means for the problem to be solved. Your shoulder hurts when you lift your arm over your head, your car has a strange clinking noise, your bonus check has not arrived in time to pay for the cruise, or you are not getting to happy hour on time. In each case there is a vision of the solution: You can lift your arm above your head without pain, the car drives quietly, the cruise is paid for and you are enjoying a Mai Tai on the forward deck, or you are at happy hour on time imbibing your favorite repast.

It is the customer or problem owner who has the vision, and they always have a vision. They do not always tell us what it is, especially if they are not asked. They may not be able to express what their vision is. Regardless, you need to know what their vision is because their vision is the embodiment of their expectations. You want to get these overall expectations early on in the process so you can deal with unrealistic expectations and modify the vision to accommodate reality early in the game, before these expectations become solidified. Never disabuse the customer of their vision initially. Take their expectations under consideration without commitment, and then get the evidence to support any reduction in expectations that must be done. The vision is the start of customer expectations

management. The vision defines for you the expectations that you can then manage.

The vision does not describe the solution or how the project is going to be done; it describes what the result looks like. It is the stakeholders' view of the solution, what they want to see in their environment when the project is done and the problem is solved. The description of how the project will achieve the product stakeholders' vision is called the project vision and it is shared among the solution-side participants. The vision is what the world looks like once the problem is solved. The vision must be expansive enough to cover the entire problem and detailed enough to pick up all the initial expectations if possible. It also provides a first cut at acceptance test criteria. The vision is the goal, the target. And, importantly, it contains many of the initial expectations that the stakeholders have of the improvements that will be made.

There are a number of reasons for establishing a vision early in the project:

- It helps to focus efforts throughout the solution development life cycle toward a common goal.
- It helps establish the acceptance criteria.
- It tells us when we're done.
- It provides a focus for each information gathering session that is conducted to define the solution.
- It verbalizes the many vague expectations the customer or problem owner has about the solution that otherwise would go unexpressed until the solution is delivered.
- It provides a target and, possibly, a metaphor for the project.
- It provides a picture or scenario for the customer or problem owner to judge whether we have a handle on the problem and solution.

Here is a description of the vision statement from a consulting company business analyst process:

The vision provides a picture or scenario for the government to judge whether we have a handle on the problem and solution, and a target for the technical team to shoot for. It also provides a first cut at acceptance test criteria. If the vision is not achieved, the solution has not worked. The vision must completely and accurately solve the problem.

Note that the vision does not state how the problem will be solved. It only states what the solution will look like when implemented.

Define the Vision

The ideal vision is a present-tense scenario that describes the problem domain when the problem is solved.

The vision is elicited from the problem owner (first choice) or other product stakeholder (as designated and authorized by the problem owner) in response to the problem statement.

The interchange usually goes something like this:

> You: Is this vendor payment terms thing the problem you want solved?

> Mary: Yes, it is.

> You: What does the solution look like? Assume the problem is solved, what is going on in accounts payable? (Or something like that.)

At this point the problem owner may become confused. She might answer, "Well, the vendor payment terms are correct and everything is working." That is a vision. It is a bit general, but it is a vision.

When you do not get a positive answer because the businessperson is unable to visualize what the world looks like when the problem is solved, you might try drawing a picture for her to complete:

> You: Let's say that the vendor database has been changed, and the voucher entry screens and other parts of the system have been updated. What is going on in A/P that is different from what is happening now?

> Mary: Well, there are no complaints from the vendors about errors in the vendor checks and the checks all get out in time to get the best terms. We are adding new vendors in minutes instead of days. We are paying the normal amount of overtime and staying within our budget. (She probably does not mention that Charley gets to happy hour on time.)

> You: Is that all?

> Mary: Well, *Financial Systems Monthly* magazine awards an annual Financial System of the Year award and I can see that with these improvements we could win the award. And, as long as we are envisioning, I can see myself as Executive of the Year at the company awards banquet in December.

The vision should be stated in the present tense as though the person is viewing the world *after* the problem has been solved and is describing what it looks like. You can orient your questions in such a way to elicit the vision in the present tense or you can record the response and rewrite it in the present tense later. The former option is more desirable, as it makes the vision come directly from the problem owner without interpretation.

Occasionally the responder is still recalcitrant. He or she simply cannot visualize that far in advance, or the problem has been so prevalent for so long that they cannot see the end of it. Perhaps they feel that by specifying a vision they will be tying themselves to a specific solution. Or they simply do not have any imagination. In this case, you suggest scenarios for them. "If we did this (describing the scenario that exists when the problem is solved) would that do it for you?" They might start offering their own alternative vision once they understand the game, or they might suggest variations to your suggested vision. Either way you have a vision and it comes from them.

Case Study

Problem: The vendor database as currently constituted does not allow for enough variance in vendor terms of payment.

Vision: The vendor payment terms are 100 percent correct. There are no complaints from the vendors about errors in the vendor checks. All the vendor payment checks are printed and mailed in time to get the best terms from the vendor. We are adding new vendor records in minutes. We are paying the normal amount of overtime and staying within our overtime budget. The AP system is awarded Financial System of the Year. Mary is designated Executive of the Year at the annual banquet in December.

Eventually the business community gets familiar and comfortable with vision statements, and you find that when you meet with a problem owner to discuss a new problem, you get a vision statement, "What we would like to see is . . . " and you end up saying, "That is a great vision, what is the problem that it solves?"

The vision may change over the course of the solution development. You may find, for example, that it is pragmatically impossible to get the payment terms to be 100 percent correct. Because it is a vision you will find it much easier to modify. The vision also has a significant effect on expectations and should be constructed with that in mind. One vice president of

agent relations in an insurance company stated in his vision, "And all my agents are smiling" (as a result of a new commission system being considered). Short of flashing jokes on their screens every 18 seconds, we were unlikely to achieve that part of the vision so it was removed.

🌐 Case Study

In our example, Mary has described two awards that she sees herself winning, but which we cannot actually guarantee with the new system. Later during elicitation and analysis we will gently disabuse her of those parts of her vision. How? Perhaps we will ask if she knows what the criteria are for winning the awards. She may say that the Executive of the Year award is subjective on the part of the President at which point she will agree that the system alone cannot ensure her of the award and we can remove it from the Vision Statement. However, note what they tell us. Mary sees the changes as so important that they could win her awards. She also clearly has a high regard for the A/P system since she considers it award winning. As a result we will certainly keep her directly in the loop throughout solution development.

When analyzing the vision statement, cross-check it with the problem statement. Make sure the problem statement is completely covered by the vision scenario. When the vision is achieved as stated, it solves the problem completely as stated. When this is not the case, revisit both statements until the vision expresses a complete solution to the business problem. Remember though, you are not specifying the solution, only that there is one, and the vision is the result of solving it. The vision:

- Defines an end target for the team.
- Provides the focus for the subsequent information gathering.
- Establishes a guideline for the acceptance tests.

When we achieve the vision as stated, we have achieved a quality solution.

Checkpoint Alpha

Because there are time and resources expended in creating the business case and/or project charter, get a signoff, at least informally, that the

organization wants the problem solved in the first place. This is Checkpoint Alpha. The cost of solving the problem may be prohibitive upon investigation, but there is no sense in exerting any more effort doing that investigation if there is no interest in solving it at any price.

When you have established the real problem, or at least what you have deduced is the real problem, you want to get answers to these three questions from the problem owner. You have a responsibility to confirm your findings with someone in authority.

1. Is this the problem you want to solve? (Confirm the problem statement.)
2. What does it look like when the problem is solved? (Obtain the vision.)
3. How will you know when you have solved the problem? Or what do you need to see to believe you have solved the problem? (Get the acceptance criteria)

Is this the problem you want to solve? It starts with a simple question: Is this a problem you want solved? There can be three possible answers:

1. "Yes, solve it under any circumstance." (When the problem must be solved regardless of cost, such as regulatory compliance or risk to organization reputation.)
2. "No, we don't want to solve it." (The response when the problem is not what the problem owner thought it was such as Alice and her blue screen of death, or things have changed since the initial problem was defined.)
3. Some form of "It depends." (Which means "we want to solve it provided it makes financial sense: the benefits are greater than the costs.")

A "No" answer means that you stop at this point and save the organization the money of doing a project that does not need to be done. You can put this effort in the success column. An "It depends" answer requires additional work on your part to define the cost/benefit or return on investment (ROI), or even the feasibility of solving the problem. We discuss this in the next chapter. A "Yes" answer, either now or after performing the financial analysis, moves us on to the next question.

How will you know it is solved? The vision describes what the solution looks like. Once you have the vision, defining the acceptance criteria is relatively easy. The acceptance criteria describe what we are going to do to prove that the problem is solved. The acceptance criteria are the answers to the third question asked of the problem owner: *How will you know that we have solved your problem?*

The acceptance criteria tells you clearly and definitively what the problem owner expects to see in the end, and sets up an agreement, if not a

---

🌐 **Case Study**

YOU: Mary, how are you going to know we solved your problem?

MARY: When I don't get any complaints from vendors. When there is no overtime for the voucher entry team. When a vendor can be added in a shorter time than now. When the vendor payments go out on time. When there are no mistakes in the payments.

YOU: Hmm. You say shorter time to add a vendor. How short a time?

MARY: Let's say two hours or less.

YOU: Okay.

MARY: Can you actually get it down to two hours? It takes six hours now.

YOU: I don't know. Let's work on that separately and time it after the screens are done. We'll set it up as a functional goal.

MARY: Sounds good.

YOU: By "on time" you mean in time to take advantage of the best terms?

MARY: Exactly.

YOU: Cool. So when we do all that, your problem is solved.

MARY: That's right. I'm a happy camper.

---

contract: "You do these things, and I'll will sign off that the problem has been solved." You can then devise ways of showing the problem owner that you have satisfied the acceptance criteria and you have the basis for the acceptance test plan.

Focus on the Problem and Vision

Once you have defined the real problem, keep the problem in front of you at all times until it has been solved and the solution is operational in the business community. The definition of the problem and subsequent focus has many benefits for the business analyst:

- The problem definition dictates to whom you will talk to gather information about the solution.
- The problem statement focuses all information-gathering sessions.
- The problem influences what the business analyst hears.
- The problem establishes the product scope circumscribing what you include of the information heard or observed.
- The form of the problem directs the choice of the modeling techniques used to describe and analyze the problem.

- The problem statement is a tool that generally resolves conflict and can be used in negotiation and mediation.
- The problem is the one unchanging part of the entire solution life cycle.
- The vision establishes the initial set of management expectations about the solution, which then can be managed throughout the solution development.
- The vision provides a target and goal for all members of the solution team to aim.

It is this focus on the problem and its eventual solution that gives the business analyst the power to perform his responsibilities and provides the impetus for the entire solution life cycle.

At the very early stages of the proposed project, in a short period of time, you have gathered some important information that will give you a head start on a successful solution:

- You have a statement of the real business problem and it has been confirmed by business management, which establishes your starting point.
- You have the vision of what the solution looks like when the problem is solved, which provides you with the initial and firm expectations of business management about the problem.
- You have the acceptance criteria, which tell you exactly what the successful solution must do when it is implemented.

The problem and vision are the primary ingredients in the product scope. Once you have established the problem and vision you can identify the constraints, risks, strategic justifications, and functional goals. These factors impact both the project and the decision to solve the problem in the first place. Let's look at how the product scope is defined.

Note

1. Dietrich Dorner, *The Logic of Failure: Recognizing and Avoiding Error in Complex Situations* (New York: Basic Books, 1997).

Define the Product Scope

First we need to understand what needs to be done. Then we decide how to do it. Then we do it. Any other sequence is asking for grief.
—Paul Oldfield

When the product includes a software development component, the product can rarely be defined clearly and completely up front. This is not generally the case in projects outside software development. When the project is to lay three miles of sidewalk, or get a new marketing brochure out the door, or redecorate the CEO's office, the products are known before the project begins. This makes defining requirements and project management much easier and more formulaic. With software development projects, most of the time we do not know what the product is, so a big part of the project is determining what the product is supposed to be.

Project and Product Scopes

Simply put, the product is what is delivered that solves the problem, and the product scope is the definition of how the product will be created. The project manager owns the project and the business analyst owns the product.

To be specific, the project scope is "the work that must be accomplished to deliver a product, service, or result with the specified features and functions."[1] The product scope is defined by "the features and functions that characterize a product, service, or result."[2] This is what will be delivered. It is the result of the project. It is the solution to the problem. The product could be an entire new system, an enhancement to an existing system, or a defect fix.
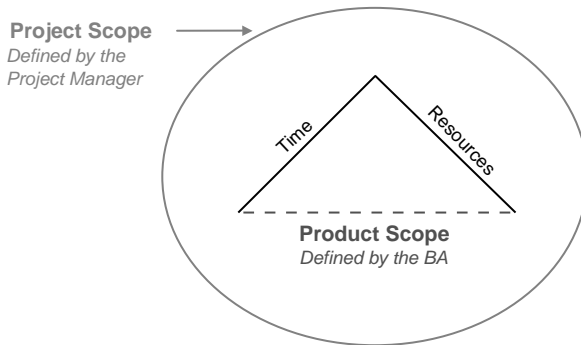
FIGURE 9.1 Difference between Project and Product Scopes

The IIBA *Business Analysis Body of Knowledge* (BABOK) describes the product scope as the "solution scope": "the set of capabilities a solution must support to meet the business need."[3] The BABOK also says, "The definition and management of the solution scope is central to business analysis, and differentiates it from project management (which is concerned with the project scope)."[4]

As shown in Figure 9.1, the product scope is the basis for determining the project scope. The project manager determines the cost and time based on the product scope, or the solution to the problem, defined by the business analyst. Doing it any other way increases the risk of project failure.

Product Scope

Part of the responsibility of defining the real problem is to gather some basic information about the problem for decision-making purposes. During the process of gathering information to define the problem, we will also gather information that provides the basis for decision making about solving the problem, which is carried forward to help guide the solution effort. This information is assembled into the product scope.

The product scope amplifies the problem statement, defining the context of the problem and solution, and answers the overall questions of who has the problem, why it needs to be solved, and what happens when it is not solved. The information, typically gathered as part of defining the problem, goes into the product scope definition.

The product scope consists of the following elements:

- The statement of the real problem that needs to be solved.
- The vision of what the world looks like when the problem is solved.

- The product stakeholders: those affected by the problem and those who may be impacted by the solution.
- The business justification for solving the problem: the benefit to the organization for solving this problem.
- The business or product constraints on the solution to the problem.
- The business risks associated with the problem and solution.
- The functional goals to be achieved by solving this problem, which are defined when the problem is large in scope or the solution is long in duration.
- The acceptance criteria, which defines how we will know the problem is solved.

Product Scope Formula

The basic elements for management decision making are defined in the product scope. They can be arranged into a formula for ease of preparation and presentation. The product scope formula looks like this:

This {solution, system, process (that we are undertaking)}
Will accomplish {these functional goals}
Resulting in this {vision}
Which will achieve these {acceptance criteria}
Solving {this problem}
Optionally the product scope may include the following
 where applicable:
On behalf of these {stakeholders, constituencies}
To realize these {justifications}
Within these {constraints}
Considering these {risks}

The information contained in the product scope is collected in the pre-project phase or early in the elicitation and makes up the core of the business case or project charter. The product scope contains all the information necessary, except for costs, for governance to determine the disposition of the project. More importantly, the product scope gives the business analyst all the information he or she needs to govern their own search for a solution to the problem.

Strategic Justification

The strategic justification for solving a problem is one of the first pieces of information you get and it comes naturally, usually without even asking.

You can usually figure that solving a problem for the sales department will generally increase revenue or market share. If not, simply asking why will usually get the answer.

The strategic justification is a statement of the basic benefit the organization gets from solving this problem, such as:

- Increase revenue.
- Decrease cost.
- Comply with regulations.
- Improve productivity or efficiency.
- Increase market share.
- Increase customer satisfaction.
- Increase shareholder value (for publicly traded firms).

This high-level business justification tends to be the same for similar industries, and the list of potential justifications remains static. It takes a short time to define a list of project justifications for your organization and, once defined, it rarely changes.

The business analysts brainstorm among themselves the justifications for the problems they have solved in the organization (or in similar organizations in which they have worked as business analysts previously). The list of all the possible justifications is placed on a whiteboard or flip chart and reviewed to remove redundant or non-applicable justifications; for example, one that is too specific to one constituency.

Once the list is created it can be copied and disseminated to all business analysts. The determination of the business justification(s) is then a matter of selecting it from the universe of valid justifications for your organization.

The strategic justification is a quick response to anyone's question, "Why are you solving this problem?" and may be tied to a corporate strategic goal for alignment purposes.

🌐 Case Study

The Justifications

You have passed through Checkpoint Alpha and gotten the problem and vision. Now you declare the justifications for the A/P effort to be decreasing cost and improving productivity. Mary, the problem owner, confirms it.

Business and Product Constraints

We can assume there will be technological constraints on possible solutions to a business problem, and those belong to the solution team. However, there may also be business or product constraints on the solution. Most of these constraints will be known up front, although some may crop up later in your investigation. For example, business constraints may be government regulations that limit the solutions we may come up with, while product constraints may be organizational policies, such as security or privacy.

The product constraints are different than the project constraints managed by the project manager. Project constraints circumscribe how the project will be executed, such as the availability of appropriately skilled resources, availability of necessary equipment and tools, impact of other ongoing projects, participation of outside forces such as vendors, and so forth. The constraints we are talking about here circumscribe the solution that is produced, and may affect the solution document or requirements, the implementation design, or the transition of the product into production.

Most business or product constraints fall into one of these few categories:

- Organizational culture.
- Laws, regulations, and external controls.
- Geographic considerations.
- Internal documented organizational policies and procedures.

The business or product constraints pose limitations on the solution, both yours and the solution team's. There may not actually be any business or product constraints for a given solution or at least that you know of at this point.

Case Study

Defining the Constraints

Playing it safe, you include a statement that the solution must comply with the current accounting practices of the corporation. Considering that the problem is in accounts payable and that will mean directly updating the general ledger and other financial reporting elements of the organization, you also include a statement that the solution must comply with any applicable provisions of the Sarbanes-Oxley Act or other related regulations.

Business and Product Risks

Almost from the start you will learn about the risks. While you are defining the problem the risks start appearing. You discover the primary risk immediately by asking, "What happens if we don't solve this problem?" This is a risk that must always be known from the start. It helps determine whether management will decide to solve the problem in the first place and provides additional motivation throughout the project. Any discussion about what might happen in the future if things keep on the way they are is a statement of risk.

The risks you capture are not limited to the constituency for whom we are solving the problem. Obviously, there is joy in the directly affected department when they are no longer in pain. You need to look further and answer other questions: What happens if we do solve the problem? What are the impacts throughout the organization of the changes we are going to implement? The executive decision maker needs to know who else is going to be impacted by solving this problem and what those impacts are, and the project manager and solution team need to know the impacts so that they may design the solution to avoid or limit those impacts. The greater the breadth of impact, the higher the risk that something will go wrong.

Some of the typical risks that we may identify at this time are:

- The ability of the organization to absorb the changes necessary to solve the problem.
- The completeness of the solution.
- The impacts on the process under consideration and any other business processes that intersect.
- The impact on the process workers.
- The risk that the solution does not achieve the expected benefits.

Case Study

Defining the Initial Product Risks

Your analysis of the accounts payable problem shows the following risks associated with not solving the problem:

- Loss of money from failure to take discounts.
- Loss of vendor relationship and cost of getting new vendors.
- Increased employee dissatisfaction by overtime work and missing happy hour, resulting in higher turnover, which result in increased HR costs for replacement of staff.
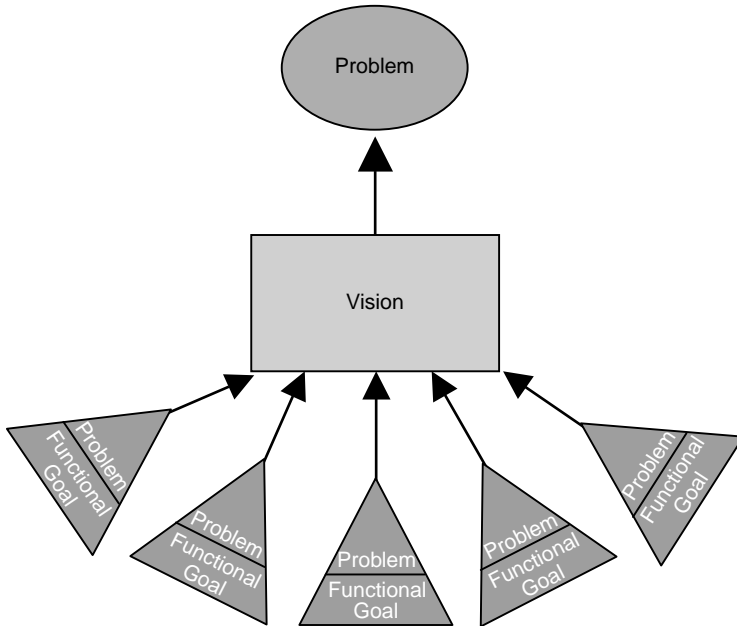
Functional Goals

Functional goals decompose the overall vision into demonstrable deliverables to the business community in order to gain mid-development feedback. When you have a larger problem to solve or one that spans business areas, you can make the job easier by breaking it into functional goals to be achieved separately. Functional goals should be unique and separately attainable. The goals should not overlap. All goals put together must attain the vision and solve the problem.

Functional goals are optional. Functional goals are not typically defined on smaller projects. The business analyst determines the functional goals during the problem definition effort or during the solution definition stage. Most of the time, the breakdown of functionality is obvious and does not take much analysis to determine. Each functional area of the problem or vision comprises a different functional goal that is recognized by the business community. And each functional goal has a deliverable that proves the functional goal has been met.

From a business perspective there typically are some clear partitions of the product to be delivered. For example, when the solution affects different constituencies and can be delivered at different times, it makes sense to deliver separate operational pieces rather than the whole product at one time. The order of delivery does not have to be logical from a technical perspective. Functional goals may be wholly dependent on the availability of the business to absorb the change. For example, if the back end posting to the general ledger has to be reviewed by accounting, and they are going to be totally absorbed by year-end closing in January, the month before the project is due, it makes sense to do the general ledger piece earlier in the project when the process workers are available for review.

Each defined functional goal solves an individual problem associated with achieving the vision and solving the problem, as shown in Figure 9.2. It is unlikely that you will be able to identify all the functional goals at the start. However, as mentioned, some are obvious. An example of a product broken into functional goals is a web site that provides a specific service to the customer. The functional goals for the web site might include:

- Creation of the content for inclusion in the Web site.
- Provision of three servers to support the Web site.
- Set up the security services to prevent unauthorized access.
- Establish an encryption scheme to ensure privacy and data integrity.
- Establish the rules for Web access and data access.
- Create the tables and databases that will be accessed.

FIGURE 9.2 Relationship of Functional Goals to the Problem Statement

The process for defining functional goals is the same as that for classic functional decomposition:

1. Start with the vision.
2. Break the vision down into individual business objectives or functional goals, which must be met to achieve the overall vision.
3. Define the problem each goal or objective solves in the accomplishment of the overall solution.
4. The vision itself may then be expressed as a series of achieved goals.
5. Verify that the complete set of functional goals, when achieved, attain the vision and solve the problem.

The functional goal statement has three components:

1. It states something to be achieved (the results).
2. It includes a measurement by which the achievement can be determined.
3. It defines the date (or timeframe) by which it will be achieved. If the goal is to be achieved before or after the project end date, it states that date.

🌐 Case Study

You might define the functional goals for the A/P effort as:

1. A list with all the possible payment terms that our vendors have offered us.
2. A revised vendor database that includes all possible vendor payment terms.
3. All vendor screens in accounts payable are changed to reflect the new vendor database.
4. Vendors are paid in a timeframe to get the maximum discount from the vendor payment terms, and so forth.
5. Reduction in the time to set up a new vendor by four hours (uncovered during Checkpoint Alpha (Chapter 8))

Political Success Factors

A critical success factor (CSF) is an element that is necessary for the solution to be successful. We define the primary critical success factors by defining the real problem, the vision, and the acceptance criteria.

However, there may be a requirement, feature, or even solution method not already included in the problem, vision, and acceptance criteria without which one or more product stakeholders will not deem the solution successful. These are called political success factors (PSFs). While you can solve the problem perfectly well without satisfying these factors, the stakeholder will not deem the result successful in his or her eyes.

The PSFs may be picked up during your problem investigation. In the example of the 50-inch screen, presented in Step 5 of the problem determination game in Chapter 8, the problem to solve is to display all the VP's reports. However, when you solve the problem without including a 50-inch screen in the solution, the VP may consider the effort a failure even when they can see all their reports.

You may encounter a stakeholder who tells you how to solve the problem. What that stakeholder may be telling you is, "Solve my problem but make sure you do this." That identifies a political success factor. You need to ask a few more questions to confirm whether the stakeholder is really committed to their solution.

To validate the PSF, ask whether the problem can be solved without it. If it cannot be solved without the requested element, then it should be included as part of the problem statement, vision, functional goals, and so on.

If the problem can be solved without the requested element, then most likely it is indeed a PSF.

In all cases with PSFs you need to make a decision whether to include the requested element in the final solution. You may decide that the solution you define is best for the organization, and you are willing to live with the political fallout. Or you may determine that the requester simply does not have enough political clout and you can ignore him. Or you may put the extra time and effort in to include the request in the solution.

Case Study

In our A/P problem, you have identified one PSF: the speed of entering voucher data that was requested by Susan, the supervisor. Suppose you satisfied all of the acceptance criteria stated by Mary and in doing so each voucher entered still takes the same amount of time to enter. You have solved the problem (vendor payments made on time, etc.) and you have not met a stated issue: speed up the data entry. Susan may not be too happy and judge the solution to be less than successful. It did not meet her expectations. You get together with Susan to determine what she means by fast and how important her request is. Should you determine that it is indeed a PSF, you make sure your solution shortens the voucher entry time, and provide measurements to prove it.

Managing PSFs is a critical part of managing expectations. Remember that the primary PSF is solving the problem. Regardless of the importance a businessperson or problem owner puts on a particular PSF, it is still negotiable. Some PSFs are simply not reasonable or feasible. You cannot implement a completely brand new business process without incurring some transitional loss of productivity. In the end, the trade-off for the stakeholder may be to solve the problem completely and correctly, or include that particular personal PSF. Determine this as early in the solution life cycle as possible so that the appropriate decisions can be made and expectations set.

Product Scope Formula

Recall the scope formula from the beginning of this chapter:

This {solution, system, process (that we are undertaking)}
Will accomplish {these functional goals}
Resulting in this {vision}

Which will pass these {acceptance criteria}
Solving {this problem}
Optionally, the product scope may include where applicable:
On behalf of these {stakeholders, constituencies}
To realize these {justifications}
Within these {business or product constraints}
Considering these {business or product risks}

The product scope can be adequately expressed with the first set of elements shown: problem statement, vision, and functional goals. The remainder of the elements comprising the product scope is optional in that they may not exist, or you may decide not to include them in a formal statement of product scope. When should you include the optional items? Here are some guidelines:

- Include the product stakeholder when the constituency is very large or politically powerful and may influence the decision to solve the problem. Otherwise, most of the time the stakeholders are obvious from the problem statement itself.
- Include the strategic justification when the problem is politically sensitive and you need to provide additional rationale to solve the problem.
- Include the constraints when there are business or product constraints. Usually there are not.
- Include the risks when they have been specifically stipulated, when there is a product deadline, or when the issue has political overtones.

Measuring

The business analyst can create the acceptance tests that show that the software solves the problem (more on that in Chapter 16). This involves establishing a form of measurement that will prove that the system continues to solve the problem, and which can be used to gauge whether the system is encountering any problems in the future. There are several reasons for constructing these measurements:

- In order to verify that the problem and conditions are as stated and not just perceived.
- In order to compare the problem to the solution to demonstrate to the organization the increased value from solving it.
- In order to have the numbers ready when there is a request for an ROI or other justification analysis.

Include the measurements in the solution so that you have baseline productivity established for future changes and a gauge to measure against to see that the problem stays solved.

Case Study

For our continuing example of the accounts payable system problem you might establish the following eight measurements immediately:

1. The time between receiving a vendor invoice and the time it is paid.
2. The number of times the payment was made in time to get best terms, and the total amount of money saved through those discounts (instead of paying in the normal 30-day cycle).
3. The number of times payments were made late and did not receive best terms, or incurred late payment penalties, and the total amount of money lost.
4. The number of vendors that have special terms that cannot be handled with the current vendor database structure.
5. The current average time to set up a new vendor.
6. The number of complaints received from vendors over a period of time.
7. Amount of overtime paid to the voucher entry team over a given period of time.
8. The number of times Charley is late for happy hour.

Note how in the case study's measurements, you expect the results to be lower with the first and third measurements after the solution is implemented and higher with the second.

Take the Technical Pulse

At some time before the business case or product scope has been approved and before the customer has authorized action to be taken on the problem, meet with the development community and let them see what is coming down the road. This is not a mandatory meeting. For one thing, there may not be a designated project team or even project manager at this point, especially if the project does not officially start until a project charter is signed. However, when the project manager or

team is known, such as when the same group does all the development for the organization, or when there are standing project teams which handle all changes to certain systems, meet with that group and show them the product scope.

The purpose of this informal meeting is threefold:

1. Get technical feedback on the problem to ascertain that the solution is feasible with the current technology available in the organization.
2. Identify any possible solution that IT has already implemented elsewhere in the organization that may be applicable.
3. Understand some of the potential technical challenges facing the solution team to help you in your dealings with both the business community and the solution team.

Bringing the technology team in early helps reduce the gap between the development community and the business community by exposing the technology team to the business side of the problem.

Even when an assigned technical team is not available, it is still a good idea to run the product scope past someone in the technical community who can provide feedback about technical feasibility. It is foolish to proceed with a problem-solving effort when any solution you come up with is going to be technologically infeasible to implement.

It does take a bit more time to develop the product scope and this may seem risky especially at the start of the project when everyone seems to want to see the developers jump into coding right away. However, the time is well invested and will reward you with a higher level of focus throughout the project. You will find that business expectations are easier to manage when you have the product scope as a reference. You will also find that dealing with the solution team becomes easier when everyone has the same goal to achieve.

Applying the Product Scope

"We can't get good requirements because the users and management don't focus on requirements."

Get the users and management's focus away from requirements and onto the real reason you are there: solving the business problem. The requirements are mere details or blueprints. Once you have established the problem and vision, keep everyone focused on those two items. Make the requirements simply the documentation of the solution. Get the solution first.
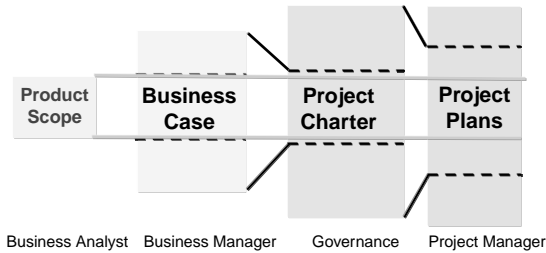
FIGURE 9.3 Progression of the Product Scope in Project Documents

As you can see from Figure 9.3, the product scope provides essential information about the problem that drives the solution effort. It provides the majority of the information the decision maker needs to determine whether to solve the problem. The information defined by the product scope is at the center of the business case, which may be prepared by the business manager or typically by the business analyst. The business case provides the central information, which makes up the project charter, and the project charter is essential to the project plan that drives the project. So the product scope is the basis for the entire project.

The problem statement, along with the vision, provides a checkpoint along the way to implementing the solution. The product scope is a framework to develop the solution in. You use the product scope:

- To circumscribe every information-gathering session that you conduct.
- As a guideline for all subsequent solution documents produced.
- As a validation and verification for each solution deliverable along the way.
- To keep everyone on the solution team and among the product stakeholders on track solving the same problem.
- As a goal to achieve that establishes the successful end of the solution life cycle.

Armed with the product scope, you can now develop a business case when such a document is required or recommended. The next chapter discusses the creation of the business case and the alignment of the solution with the organizational strategies and mission. When you don't have a requirement to create or define a business case, or when the project has already started, the product scope provides the high-level guide to gathering the information to define the problem domain and the solution. Information-gathering techniques are discussed in Chapter 11 and defining the problem domain is described in Chapter 12.

Notes

1. Project Management Institute, *The Guide to the Project Management Body of Knowledge, 4th Edition* (King of Prussia, PA: Project Management Institute, 2008), 103.

2. Ibid.

3. International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge*, version 2.0 (March 31, 2009), 232.

4. Ibid., 8.

CHAPTER **10**

Confirm Alignment and Financial Justification

You don't have to agree, you just have to be aligned.

—Corporate policy

Determining the alignment of a specific project with the overall organizational strategies takes a business analyst out of the comfort zone of documenting requirements for IT and places him directly in the realm of senior management. Some business analysts may be uncomfortable in that rarified atmosphere. However, the business analyst role has a responsibility to perform due diligence to ensure that the proposed problem and associated solution are aligned with the current organization goals, strategies, and operations. This means that to be successful, a business analyst must have knowledge of the organization mission, goals, and long-term strategies. The concept of being the guardian of corporate alignment is a frightening one for most business analysts. It conjures up images of politics, summary trials ending in dismissal, and accusations of disloyalty and insubordination. Fortunately many organizations have not realized the benefits a business analyst brings to the alignment evaluation process and have not required their business analysts to perform such an evaluation. The organization just blames the business analyst when the project or product varies from the corporate line.

The Business Case

"What's the best way to create the business case? Is it the job of the business analyst?"

In organizations that take their problem solving seriously enough to make an informed decision through an established process, decision making is based on some document that establishes what the problem is, what the solution will be, and what it will cost in terms of resources and time. This document may be called a business case, a scope and vision document, a product scope, a project charter, or other title based on the applicable standard or process in use. All of these documents serve the same purpose: to provide the information necessary for someone to make a decision to solve this business problem now, later, or never. The overall goal of the business case is to minimize risk in decision making.

According to PMI's PMBOK, "The business case or similar document provides the necessary information from a business standpoint to determine whether or not the project is worth the required investment. Typically the business need and the cost-benefit analysis are contained in the business case to justify the project."[1] The BABOK says, "The Business Case describes the justification for the project in terms of the value to be added to the business as a result of the deployed solution, as compared to the cost to develop and operate the solution."[2] The business case may simply state the rationale for solving a business problem, or may go into great financial detail to support the solution.

The business case is prepared before the project is approved. Since there is no approved project, and therefore no assigned project manager at the time the business case is prepared, just who prepares this essential document? The PMBOK says, "The requesting organization or customer, in the case of external projects, may write the business case."[3] The BABOK is clear about who writes the business case: "The BA collaborates with subject matter experts (the business sponsor, business representative(s) and IT management) to scope the proposed project, make time and cost estimates, quantify business benefits and prepare the business case."[4] While the two appear to differ, the PMBOK does not say that the business sponsor actually *writes* the document. It is unlikely that an upper-level manager who is acting as a business sponsor will take the time to research and document the business case. The business analyst does that and the upper-level manager will sign it and issue it.

The Value of IT

An IT project is only worth the value it brings to the business. To gauge the value of solving a problem or creating a product, you need some background knowledge. For example, you need to know:

- The organization's external environment: the market(s) the organization is in, their competitive positions, current state of the industry, legal and regulatory constraints, and so on.
- The organization's internal environment: its financial position, organizational culture, the relationship of IT to the business, the coordination and integration of all the organization's entities, and so on.
- The business environment: the business processes, the demands of the organization's customers, the relationships among the business units, partners, and supply chain factors.
- And, most importantly, what the organization considers valuable: the mission, the goals, and objectives of the organization and the strategies to achieve them.

Some problems do not need to be solved and some problems are not problems at all; they are simply reflections of political expediency. There are many projects brought to IT that should never have left the executive office in which the idea for the project was hatched. There are problems that should not be solved because the solution will cost more than the problem. There are times when IT is directed to implement a solution for which there is no real problem.

In all these cases, when the business requests one of these projects, IT typically complies with the assignment, only resisting when there are schedule or resource issues. IT does not challenge the motivation behind the change. As a service organization, it is not IT's job to do so.

The advent of the PMO in any of its incarnations, or any other governance body, provides an intermediary unit to validate the change requests and to prioritize them. The business analyst typically works with the PMO to provide the information for vetting. In the absence of a governing body, the business analyst has the responsibility of performing the due diligence, however, the business analyst does not make the decisions; he only gathers the information on which the decisions are based.

You must be relentless in your investigation to determine that the problem and solution are both aligned with the organization's business drivers, strategies, and goals by evaluating the alignment of each proposed project. The solution to the business problem should align with the mission and strategic goals of the department requesting the solution, and, of course, with the mission and strategic goals of the overall organization.

Considering Alignment

Whether you are in an organization that considers IT a back-end function existing solely to support the business or as an equal partner in the furtherance of the organization's goals, every IT project must have a business

FIGURE 10.1 Alignment Hierarchy

justification. The work that the business analyst does produces value aligned with some organizational strategy or tactical effort so it achieves at least part of an organizational goal.

More than alignment, you can help the organization synchronize IT with the rest of the business so that IT and the business make decisions together. Also, you can bring IT into a position with the business to help shape rather than just enable the organization's strategic choices.

There is a hierarchy at play within the organization forming a pyramid-like structure, as shown in Figure 10.1, around which the organization is based. Enterprise strategies start with the organization mission and its associated goals.

Organization Mission

Mission statements serve organizations by giving direction and clarifying choices. Organizations can have integrity by defining their values and principles and then making choices that are in line with them.

—Stephen Covey

Every organization has a mission. An organization's mission statements can be found on the organization's home Web page, on laminated business card–sized reminders employees can carry with them at all times, on posters hung throughout the organization's headquarters, on the organization's annual report, stationery, advertising, marketing brochures, and so on. The

mission is the organization's reason for being—the organization's unifying overall purpose.

Here are some examples of organization missions:

- At Microsoft, our mission and values are to help people and businesses throughout the world realize their full potential.
- Google's mission is to organize the world's information and make it universally accessible and useful.
- The Department of Homeland Security's overriding and urgent mission is to lead the unified national effort to secure the country and preserve our freedoms—United States Department of Homeland Security.
- To be the leading global provider of networked consumer electronics and entertainment—Sony Corporation.
- McDonald's vision is to be the world's best quick service restaurant experience. Being the best means providing outstanding quality, service, cleanliness, and value, so that we make every customer in every restaurant smile.

In the end, it is not just the mission statement, but how it is understood and acted upon within the organization.

The mission of a fictional financial products and services company might be:

To be the primary provider of financial instruments that secure the health and livelihood of our customers and promote wealth throughout their lives.

The business analyst who can directly link a solution effort to the organization's mission has little political trouble with support for the solution. The opposite is true as well. For example, when someone at McDonald's suggests that the company add a new line of cook-to-order Beef Wellington (takes 45 minutes to an hour to prepare) the business analyst could point out that as tempting as it might be, the project is contrary to the mission of best quick service restaurant. Similarly, outfitting the restaurants with wall-mounted television sets in view of the checkout counters tuned to the news or a drama gives the impression that the wait is going to be long enough for patrons to need a television as a distraction. In a bank, where patrons want exacting accuracy and understand that a single transaction may take a while, having a television screen to watch as a distraction while they wait may be a good idea. The business analyst checks every proposed problem against the mission to challenge any problems whose solution does not further the organization's mission.

🔍 **Example**

A training company had as its mission to be the leading purveyor of classroom-led technical education in the world. It set as one of its goals to build 20 classroom facilities in each of six major metropolitan technology centers in the world in which to hold classes open to the public. When the market for technological training dropped after the dot-com bust, the company changed its goals. The company decided to opt for a larger number of smaller classroom facilities and increase focus on on-site training.

Organization Goals

Organization goals may relate to increasing customer value, making business operations that drive value to the customer more efficient. This increases value for the shareholders, and improves the capabilities of human resources and other corporate assets. Goals for an organization can change as the business and economic environment changes. Goals should always be aligned with the corporate mission. Each problem that the business analyst addresses should have at least one corporate goal with which to align.

A corporate goal from the fictional financial company mentioned earlier might be "Increase sales by 20 percent within the next two years."

Organization Strategies

The strategies describe how the goal is achieved. The training company in the example sidebar started with a strategy of opening the public classroom facilities in the locations of largest demand, and then working down the list. The sites were to have full printing facilities and everything else to support the classes delivered on that site. When the goals changed, the strategy changed. The company made arrangements to ship the classroom materials from several geographically convenient centers.

Going back to our financial company, here are some of the strategies that the upper-level management of our fictional financial company decided on to achieve the previously stated goal:

- Retire poor performing product lines to focus sales and support on the product lines that have better return.

- Pay commissions on sales rather than on premiums or full payment to provide better incentive to the sales people.
- Redistribute the sales territories to maximize the performance of the top sales people and increase the sales of the lower performing sales people.
- Add a new product line each year to fill in gaps in the overall product portfolio.

Department-Level Mission, Goals, and Strategies

As shown in Figure 10.1, each department in the organization has its own mission, goals, and strategies. The three departments (HR, Marketing, and IT) all have their own missions to achieve as part of the overall organizational mission. The alignment of a department's mission, goals, and strategies with the overall organization's mission, goals, and strategies is not the business analyst's responsibility. The business analyst may help departmental management with strategic planning that aligns with the organizational goals.

Example

Department Cross Purposes

Sometimes one department runs a project that makes sense for the department and, by extension, the organization but which has a negative impact for another department.

A dozen years ago I was working for an IT department that supported both Microsoft systems (Windows 3.1 and 95 at the time) and Apple Macintosh computers. The graphics department used the Macintosh computers. To save money and standardize maintenance and support operations, the IT Department decided to replace all Macintosh computers with Microsoft systems. By doing this, IT planned on saving the costs of supporting two different computer systems. IT had done a study and concluded that the software currently in use in the graphics department was available on the Windows system or there was an acceptable off-the-shelf software alternative. The graphics department was not too happy because they felt that the graphics software at the time was much better on the Macintosh than Windows and produced better results. However, IT owned the computers so graphics had to go along. Several months later the frustrated graphics

department started purchasing Macintosh computers out of its own budget. The issue reached the executive suite when the Annual Report and some of the critical marketing pieces supporting a new product were less than stellar and the graphics department blamed the lack of quality on the PCs. Management required IT to return the Macintosh computers to graphics because the quality of the graphics output had deteriorated. IT had to replace the new PCs with the latest Macintosh computers and associated graphics software incurring additional cost.

Had there been a business analyst associated with the project, the business analyst might have recorded the graphics department's objections, obtained samples of work done on both computers, and presented the results to management to determine if changing the computers was worthwhile before the change was made.

At our fictional financial company various departments are involved in achieving the stated goal. The corporate strategies govern the departmental mission, goals, and strategies. For example:

- The product development department has the goal of adding two new product lines over the next two years.
- The sales department has the goal of defining the strategy for the new commission structure for the sales people based on paying commissions sooner and applying charge-backs for after sale changes. Sales also has to reorganize the sales territories.
- The payroll department's goal is to come up with a strategy to incorporate the new sales commission structure into the payroll process including changes to the payroll computer system, pay forms (that now have to include the charge-back information), and end of year process (how do they handle charge-backs from one tax year to the next?).
- Marketing has a goal to create the marketing strategies for the new product lines and have the announcements, advertising, Web content, and so forth prepared in time for product development to roll out the new products. Marketing also has a goal of removing non-producing product lines based on their marketing and sales information and providing the appropriate materials to the sales persons for customer product conversion and to the customers to make the changes in the product lines palatable.

Each of these department strategies is going to require one or more projects to realize the departmental goals, and most of those projects are either IT-based, or have a significant IT component. The projects are all tactical.

At the Tactical Level

Business analysts make sure a proposed project is aligned with the organizational and department strategies and goals at the tactical level. To understand the process of determining the alignment, let us look only at the single goal of our fictional financial company: to increase sales by 20 percent in two years. Table 10.1 lists a number of proposed projects from all areas of

TABLE 10.1 Project Alignment Examples

Project Proposal	Source	Alignment
Create a software program that relates sales support personnel with overall sales to identify poorly performing sales support personnel.	Vice President Sales	NOT ALIGNED The result does not necessarily increase sales, and is not part of any of the strategies, especially the sales department.
Create a computer function that shows product line performance in a graphical form over any selected five-year period.	Marketing Department	ALIGNED Identifies the poor performing product lines for elimination.
Change the territory assignments in the databases and systems.	Sales Department	ALIGNED
Upgrade the accounts payable system to correct invalid vendor payments.	Manager of Accounting	NOT ALIGNED It is a project that should be done; it is just not aligned with this goal.
Tie the competency metric system in training with the performance and evaluation system.	Human Resources	NOT ALIGNED Even though it sounds like a way to improve the sales force, it is not directly aligned.
Revise the commission system to include the charge back structure.	Payroll	ALIGNED
Change the existing fulfillment processes to accommodate the new product lines.	Operations	ALIGNED

the company. Some of the proposals may be aligned with other corporate goals; however, for the example we are only considering this one goal.

In the last proposal the business analyst might suggest that the fulfillment process be made more generic. This allows operations to add new product lines without having to make changes to the existing software or databases, which clearly achieves the overall corporate goal of increasing sales. While this approach probably costs more and may take longer than making the new product line changes each year, should the corporation not plan to continue to add product lines, a generic fulfillment process would not be considered within alignment.

Determining the Value of the IT Project

Many organizations run technology projects because the technology is cool, or more modern than what they had before. I have witnessed, as I am sure you have as well, a technical solution brought up by IT which then looks for a problem to solve with it.

Not every proposed initiative is based 100 percent on business. Many times a business initiative is influenced by the availability of a software package or other technology. The proposal may be for something the business has wanted for a long time and been unable to accomplish technologically. However, to have real value, each IT project must provide some contribution to the business of the organization. When it doesn't, the business analyst should challenge its alignment.

The Eternal Project

The issue of alignment does not end once the project charter is approved and the solution team starts its work. The project must be as aligned with the organization's goals at the end as it is in the beginning. Organizations change goals and missions; they get bought or change shape or move into different markets. The conditions that caused the problem the project was created to solve might change or go away. You might be in the middle of creating the new customer relationship management (CRM) system when the company buys another company that already has a CRM system that is applicable, rendering the project you are on redundant.

Projects can take on a life of their own sometimes, continuing even when the product is no longer needed. The project might continue in a maintenance mode, providing additional features and functions that are not really needed to solve a problem, but serve the purpose of giving the project team something to do so that the organization has technicians experienced with the system in case a new problem develops.

This happens when no one alters the goal of the project or acknowledges that the goal has changed in light of current business circumstances. In other words, the problem that the project was to solve has been eliminated by changes in conditions. Driven by the initial objectives of the project, IT management and the project manager continue to march until the business tells them to stop. The business may be distracted by other issues, or there may be an ineffective transfer of power when business management changes. The project continues indefinitely, attempting to solve a problem that has already been solved or no longer needs solving.

Some projects are subject to what is called sunk-cost bias. This is when management continues to fund an inefficient project so as to not waste the money already invested into it, in other words, throwing good money after bad.

Retain your independent and objective outlook throughout the project. In this way, you can avoid the emotional pull that projects can exert on the project team that make it difficult to realize that the results of the project are no longer necessary or that the objectives have changed. Consider only the costs and benefits of completing the project.

Killing a project should not carry an aura of failure with it. When you can kill an unworthy project in the inception phase before a lot of money has been spent, this is a good thing and provides value to the organization. It saves the organization not only the money, but also the political problems that come with a project that has been unwisely approved.

Political Issues

Nearly all projects that bring about significant change to the organization have political overtones. There is always someone who is against the change and against the project. There are those who would like to see allocated funds diverted to solve one of their problems. There are those who are not involved and feel they should be, and those who are involved and prefer not to be.

When there is any question about whether the project is correctly aligned with corporate strategies and goals, the project could be in jeopardy. This puts you in the difficult situation of trying to promote a change which has political enemies and potentially no support from upper-level management.

Whenever there is a political challenge to your solution, simply point out the alignment and the challenge usually ends there. Should the politician wish to challenge the project, he must challenge the corporate goals or strategies and that requires a discussion with those in power. When the project is clearly aligned with corporate goals there is little argument, and the politics flow elsewhere, not toward you.

"We knew it wasn't a good thing to do, especially from a cost perspective, but it was [the boss's] pet project and he believed it was the greatest thing going. Then everyone got blamed, especially us. How can you deal with something like that?"

It Is Hard to Buck the Boss

When you report to the business unit or are assigned by IT to the business unit, you will find it difficult to inform the unit manager that one of her initiatives may not be aligned with current strategic plans. The obvious ploy is to ignore the issue and let someone else, preferably someone of higher authority, break the news to the manager.

The safest way to do this is to ask questions. Devise questions that clarify the request for work from IT. Even if you are absolutely clear about what is going on, you still want to question the rationale for the request. Use questions that start with:

- "I am not quite sure I understand. Can you clarify . . . ?"
- "I am probably not seeing this from the right perspective. . . . "
- "I am not sure I can explain this request clearly to Jack, the project manager. Can you explain it again?"
- "The CIO has been challenging projects that are not based on standard equipment. How can we present this so that the CIO will not object?" And so forth.

Your goal in this interchange is to let the manager see for herself that there is a problem with the request and either clear it up or drop the request entirely.

Taking the time to confirm the alignment of a project or to expose a misalignment brings value to the organization and is the business analyst's responsibility. Sometimes the simple act of tactful questioning can move a project toward a more productive conclusion.

Even when nothing changes, that moment of project introspection gives everyone involved an increased feeling of confidence that they are producing something of value for the organization.

Provide Financial Justification for Solving the Problem

Although many organizations have a department devoted to performing financial analysis—such as return on investment (ROI)—in a number of organizations providing financial and other justification for solving the problem is the business analyst's responsibility. How the justification is done is

another question. Some companies rely on the return on investment analysis as the primary determinant of whether a problem should be solved. Others prefer cost/benefit analysis and/or feasibility studies. Business analysts either provide the necessary information to the accounting types who put the numbers together or they complete the spreadsheet themselves.

> *Our job as engineers is to conceive and develop technological products that serve the needs of humanity or—let's be honest here—that sell in large volume.*
>
> —Bob Colwell, former CTO of Intel

While it may seem obvious to the business analyst, the champion, and to the problem owner that the problem needs to be solved, or that improving a business process will increase the value of the process to the organization, upper-level management needs a financial justification for solving the problem. The business analyst needs to answer the following question: How does solving this problem benefit the organization financially?

The business case provides the project sponsor, or governance committee, information to decide whether to solve a given problem. The role of the business analyst prepares the business case or similar document by:

- Defining and including the product scope in the document.
- Identifying and quantifying the benefits of the solution to the business community.
- Identifying and quantifying the costs as estimated by the solution team.
- Defining a measurement and process for measuring for both the costs and benefits.
- Assembling the information into a readable document for evaluation.

"Is it necessary to provide cost justification such as ROI for projects and, if so, how do you do it?"

Return on Investment

I have found that only a minority of business analysts regularly perform ROI analyses. Many have never done it and some have never even heard of the practice. The BABOK suggests that an ROI analysis is part of the project scope document and prepared primarily for "sponsors and other managers at the executive level."[5] However, when policy or upper-level management requires the ROI, the business analyst is usually called upon to provide the information for the analysis even when actually engaged in running the numbers.

When done, the results of an ROI analysis should be included in a deci-sion-making document. It should provide information to management for their review that helps the decision makers make the appropriate decision to solve the problem or not.

You are typically able to measure the investment needed to develop the solution in dollars. You may not be able to translate the return or benefits of solving the problem into dollars as easily. For example, there is a high customer demand for better navigation on the primary com-pany Web site, as determined by marketing calls. It will take eight hours for a user experience analyst to determine the new navigation scheme and 24 hours for the developers to make the changes. Add another eight hours for testing and you have a week's worth of work. But how can you determine a measurable benefit of a new navigation scheme on the web site? Will new customers come to the Web site because of new navigation? Will existing customers buy more? Or will they not like the new feature because they were used to the cumbersome old method and now they will go elsewhere?

The product scope discussed in Chapter 9 generally represents the ben-efit side of cost/benefit analysis or ROI analysis. The cost side cannot be adequately computed by IT until there is at least a product scope. The feasi-bility study also requires a product scope.

The ROI measures are set by generally accepted or legally mandated accounting standards and practices. The costs and savings or revenue might be projected over a multiyear time span to show a payback period or to estimate the present value of future returns. The business analyst's job is to determine not just how to measure a financial benefit to the organization, but also when a measurement makes sense.

To provide adequate measurements of the status quo and the expected solution, you need to establish a control group to measure before and after. This is not easy. For example, suppose you need to determine the ROI of paying for certifications for technicians. To really measure it, you would have to measure and compare a group of certified technicians against a group of noncertified technicians at the same time. If they know they are being measured it will skew the results. How do you factor out the differ-ences in skill levels on the two teams so that the difference in performance is due solely to the certification?

Some things cannot be measured. The benefits of some solutions are intangible. An intangible benefit is a benefit that does not have a direct monetary measurement. For example, a new user interface for the accounts payable system would improve the morale of the process workers. We may be absolutely sure that the morale has been increased, through satisfaction surveys, employee meetings, and so forth, but there is no direct correlation between increased morale and the bottom line.

Improved morale is an intangible benefit. When you show that the turnover rate has been reduced in the accounts payable department and that results in a 15 percent drop in hiring costs, you show a tangible benefit. Sometimes, you have to play investigator and analyst to unearth measurements for intangibles.

At this point in our problem solving, we only know we have a problem to solve. Even when we determine the benefits in present value dollars, it is going to be very difficult to coerce a firm estimate out of IT for producing an unknown solution. There may not even be an assigned solution team or project manager at this point to make the cost estimate and we certainly do not want an estimate from someone who is not going to be doing the work. On top of that, IT knows that the word estimate, no matter how preliminary it might be, and how many conditions are applied to it, usually ends up being the final budget allocation for the project.

Unfortunately, there are times when it seems that you are creating the ROI only for the sake of creating the ROI. Often in my career I have been assigned the job of producing an ROI analysis after a project is already in progress. In these cases, it apparently did not matter what the results of the ROI analysis turned out to be. In one case, the project was nearly completed when the project manager asked me to complete an ROI analysis on the project because corporate policy required ROI documentation with every project. I have also been asked to adjust the results of an ROI analysis that did not achieve advertised expectations. This is not skullduggery or cheating. The ROI can be improved by changing some assumptions on which the ROI is based.

Cost/Benefit Analysis

More business analysts perform the less rigorous cost/benefit analysis (CBA) than the ROI. Many of the business case templates in use include a cost/benefit analysis component to show what the problem solution is worth to the customer and organization.

The CBA is just as much a decision paper as the ROI and can be used where there is not an actual investment, only a cost, such as when deciding to hire a new sales person, or to adopt a new regulation such as noise abatement at airports. All the considerations of costs, investment, and tangible and intangible benefits discussed earlier apply to CBA as well as ROI. Many organizations tend to use a CBA in place of an ROI because it is perceived to be easier and less rigorous. From the business analyst perspective, both decision documents are equal in terms of the investigation that must be performed to gather the necessary information.

Proof of Solution: Feasibility Study

For small, relatively straightforward efforts, the solution approach can be determined by the business analyst alone or with a small team of experts examining the approaches in an informal working session. For larger change initiatives requiring significant investment, a more formal feasibility study may assist with determining the most viable solution option.[6]

Generally, as business analysts, when we address feasibility dealing with an IT solution, we are talking of technological feasibility—one of many types of feasibility to be studied, such as:

- Economic feasibility—can we make the change for a price we can afford?
- Business feasibility—is it within our strategic direction?
- Cultural feasibility—can the organization accept the change?
- Legal feasibility—does it stay within all our regulatory constraints?

In many cases, feasibility and understandability go hand-in-hand. A product may be considered infeasible because it is not understood. When we do not understand what the product is all about and what problem it will solve, it tends to appear infeasible on a number of levels. Your job as business analyst is to reduce complexity as much as possible during the decision making process and all through the solution implementation. When you define the business problem and vision first, you simplify the issues for all stakeholders, making it easier for upper-level management to make decisions, making it easier for the product stakeholders to visualize business solutions to the problem, and making it easier for the solution team to implement the solution.

Feasibility studies might be done as separate, independent projects. The feasibility study may be used as part of the decision-making process, or to evaluate approaches to solve the problem, especially when there are multiple solutions known or proposed during solution development.

The Metrics Game

There is a risk that the project will be successful and produce the desired product, but the benefits derived from that product do not meet the financial expectations. The project manager, who has a project budget and establishes cost controls and monitoring that are linked to

milestones, monitors the cost side of the CBA. However, the project manager does not evaluate the benefits to the organization after the solution is in place. This is your job.

In carpentry there is a rule: Measure twice, cut once. The rule says that the carpenter should measure the wood he is about to cut and then measure it again. The reason is simple: to save wood. If the initial measurement is off by even a fraction and does not fit exactly, the wood is scrapped.

The business analyst has the same rule, although it is not applied in the same way. The business analyst measures the current situation, the problem domain (measure once). Then the business analyst defines a change to the problem domain that the solution team is going to implement (cut). Once the solution has been implemented, the business analyst measures again to confirm that the solution has resolved the problem (measure twice).

To prove that the problem has been solved and that there has been an increase in value to the organization by solving the problem, we need to measure both the problem and the solution.

Identify measurements that can be taken before and after the solution is implemented that will prove that there is benefit and what the benefit is. Your primary concern is not to prove that you have performed a good ROI analysis or CBA. Your primary concern is to demonstrate to upper-level management that it was worthwhile to solve the problem. You should be able to build the measuring mechanism into the solution so that the system measures itself.

In the End . . .

Present the business case as objectively as possible and not weighted in favor of a specific solution that an upper-level manager has identified. Present the risks of not solving the problem clearly and completely, and the impact on the organization of each potential solution. When you start with the product scope you have the basic information around which to build the business case. When you incorporate Checkpoint Alpha into your problem definition process, you may find that the majority of the time you don't need to do a financial analysis.

Once you have established the problem and the parameters of the problem, and gotten agreement that the business needs to have the problem solved, you can move confidently on to the solution. Part Four discusses the process for defining the business solution to the business problem.

Notes

1. Project Management Institute, *The Guide to the Project Management Body of Knowledge, 4th Edition* (King of Prussia, PA: Project Management Institute, 2008), 75.

2. International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge*, version 2.0 (March 31, 2009), 94.

3. Project Management Institute, *The Guide to the Project Management Body of Knowledge*, 75.

4. International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge* version 1.6 (2006), 19.

5. International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge*, 77.

6. Ibid., 90.

PART IV

The Process

Once the problem has been defined and confirmed with the business community, specifically the problem owner, it's time to define the solution. To many business analysts the process of defining the solution, or identifying the requirements that make up the solution, is the primary part of their day-to-day occupation. Business analysts and requirements are often synonymous.

Communication is central to defining the best solution to the business problem. The investigation process is all about communication: asking questions, conducting interviews, and moderating meetings. The information gathered during the investigation is used to define the problem domain to give you a starting point to solve the problem. A primary purpose of analysis is to generate more questions that are additional communication and investigation. The rendering of the solution into a solution document in some form is also a form of communication and analysis, and will usually generate even more questions, resulting in more investigation. Considering the criticality of information gathering, this section starts with a discussion of the investigation process and addresses the issues of asking the right questions, getting quality information, and handling the problems inherent in human interaction (Chapter 11).

Chapter 12 presents a structured process to define the problem domain and the specific business process that contains the problem. Defining the problem domain is an essential first step for defining the solution and addressing all the impacts of that solution.

Using the problem domain and the information you have gathered, you analyze the information to determine a solution. This process is iterative. Analysis will generate questions to eliminate inconsistencies, fill holes in the information, solidify vagueness, remove redundancy, resolve ambiguities,

and identify assumptions. In each case, more information is required to complete the analysis, which means more investigation, which produces more information, which then has to be analyzed requiring more information, and so on. The analysis process is described in Chapter 13.

Eventually the analysis identifies the solution that is then documented into the solution document. This process is explained in Chapter 14.

At the end of these processes you have defined the best solution to the business problem and you can turn that solution over to the solution team to implement.

CHAPTER **11**

Gather the Information

If they can get you asking the wrong questions, they don't have to worry about the answers.

—Thomas Pynchon

During investigation, the business analyst focuses on the collection of information rather than the identification of specific user or stakeholder requirements. In this way, the business analyst can discover other requirements that might not be explicitly voiced by the stakeholder. The definition of what needs to be done to solve the problem occurs during analysis.

The classic approach to defining the solution is called "elicitation."[1] According to Merriam-Webster dictionary, elicitation means "the act of bringing or drawing out (something latent)" or "to call forth or draw out (as information or a response)." In other words, elicitation implies extracting information from the process workers and other stakeholders, not simply recording any requirements they may have.

Instead of thinking elicitation, consider the process of interacting with the business community and the product stakeholders as *investigation*. According to Dictionary.com, to investigate means: "to examine, study, or inquire into systematically; search or examine into the particulars of; examine in detail," or "to search out and examine the particulars of in an attempt to learn the facts about something hidden, unique, or complex."

The purpose of investigation is not to extract requirements from the user or stakeholder but to collect information relevant to the problem or the solution. This information is then analyzed to produce the solution. Thus the business analyst can identify associated or dependent requirements, nonfunctional requirements, and certain system requirements that the user or stakeholder does not have the knowledge or experience to identify.

Why We Cannot Define Good Requirements

Years ago I was in a meeting in Austin, Texas, discussing the requirements we had for a system under development. Actually we were complaining about the users and stakeholders who gave us the requirements and how the requirements they gave us were so bad. The users and stakeholders were of course complaining about the system we were developing based on their requirements.

At one point, I suggested that we write down all the obstacles to getting the good requirements—everything that prevented us from defining the system in such a way to prevent user complaints. We compiled a fairly long list. It was somewhat illustrative, if not eye-opening. I then suggested we evaluate all the reasons and determine what might be done to counteract the perceived obstacles that we had written down, figuring that most of the obstacles were vested in the user population and we'd have to call management in to intervene. It turned out on analysis that most of the obstacles rested with us, the requirements definers. Since that time I have asked the same question hundreds of times in meetings, conferences, and classrooms all over the world. The answers have been fairly consistent.

The following is a partial list of those obstacles as voiced over the years in these numerous brainstorming sessions with business analysts. You probably recognize some of these obstacles and have complained about them yourself. I have included a larger list in Appendix C. Take a look at both lists and determine which obstacles are your responsibility as business analyst to overcome, and which are not the business analyst's responsibility:

User availability— not getting to the right people.	Business does not want to do requirements.
Not enough money.	Do not listen to the business owners.
They do not know the problem.	Problem not completely defined.
Inability to communicate.	Scope not defined.
Focus on solution and not problem.	Lack of visual aids.
Assumptions.	Lack of management support.
Unrealistic expectations.	Politics.
Users and management give us solutions not requirements.	Getting requirements from just one person.
The BAs don't know the business.	Users do not tell us everything.
Combining tech and non-tech vocabulary.	Not knowing if you have captured everything.
Business analysts do not ask the right questions.	Businesspeople do not know what to tell us.
They do not know what a good requirement is.	Misunderstandings and misinterpretations.

User's cynical attitude in light of past system failures.

Users resistant to change.

Changing users/customers.

Hidden agendas.

Not enough questions.

Lack of technically sophisticated users.

Users do not know what they want.

The users and managers cannot agree on what the requirements are supposed to be.

Lack of understanding of environment.

No input from the user.

Improper levels of expertise.

Miscommunication.

Lack of client knowledge.

Having an open-ended scope.

Overoptimism by everyone but developers.

Lack of commitment and support from sponsor.

Poor user's skills to use system.

Quality assurance not involved soon enough.

Stakeholders forced to think in engineering terms.

Users want everything, just in case.

Users do not know what they want.

Project scope not accurate (do not understand why).

Do not know if users will use it correctly.

Deal with the least valuable person.

Jump to conclusions.

Changing business environment.

Terminology.

Poor buy-in.

Requirements are too general.

Not specifying business rules.

Scope creep.

Communication inconsistency.

Non-technical users.

Not defining terms upfront.

Problem not fully defined.

Users do not read the documents—they just sign off.

Lack of consensus on the customer side.

Lack of implementation training.

People are confused about requirements.

Uninformed management must make decisions about requirements.

Lack of commitment from business.

Not enough time.

The answer is simple: All of the obstacles are the business analyst's responsibility to remove. The business analyst can overcome each of the obstacles listed on these charts. I have annotated the list in Appendix C to correspond to sections of the book where I mention tips or techniques to help you overcome the obstacles.

Stop Gathering Requirements

Many of the obstacles on these lists arise from one common misconception held by most business analysts. This misconception is that users have the requirements, and the business analyst simply has to gather them. Many business analysts say that the primary activity of their job is to gather

requirements. After all, that is what management tells us to do: go gather the requirements.

Let us consider the directive to gather the requirements and picture what the process might look like:

> The business analyst grabs a basket or some suitable container in which to place what is gathered. The BA wanders over to the business community and asks each product stakeholder for his or her requirements which he places in the basket, like apples or eggs. Of course, since it is unlikely the members of the business community will have the requirements written down, the business analyst faithfully and accurately records their list of requirements. The business analyst collects all the requirements, brings them back to his cubicle, and transcribes them as carefully as possible, recording them word for word, into the organization-prescribed template—a requirements document. The business analyst's sole analytical effort might be to remove redundancy when two or more product stakeholders define the same requirement. Then the BA takes the requirements document back to the product stakeholders to make sure the requirements have been transcribed correctly. The stakeholders may add some new requirements or change some of those they had previously given, which the BA dutifully transcribes. Finally, the BA takes the requirements document to an authority to get the document approved. Having obtained the approval, the BA turns the requirements document over to the solution team and the job is done.

This scenario may describe a requirements documenter or requirements recorder but certainly not a business analyst. There are two major problems with this scenario. First of all, there is no analysis performed and analysis is what we business analysts do. Analyst is our last name. In this scenario there is no analysis. Is it any wonder I hear business analysts all over the world complaining about getting no respect from the developers and even less from the stakeholders? I think business analysts have considerably more to offer the organization than simply recording requirements. The only way you can elevate yourself from being a requirements recorder is to stop gathering requirements.

The second problem with the scenario is much more insidious. It assumes that the requirements are already there, and have already been defined by someone. If you are going to gather requirements, they must already exist. That means the users have to already have the requirements, even when they do not know they do; and our job as business analysts is to ferret them out, by coercing, extracting, or forcing the users to divulge the

requirements to us. For the gathering requirements scenario to work, we have to assume that the users can completely identify their business problem and come up with the appropriate solution to that problem which they then can relay to us. And further, the users can completely and unambiguously state the requirements that implement the solution in a way that the solution team can understand.

Users Do Not Have Requirements

The job of the process worker or user is *not* to define requirements for us. The job of the users of a system is to sell product, book orders, enter payables vouchers, or produce the payroll. Even if they were assigned by their manager to take a few weeks off the production line and write up some requirements, they are not trained or skilled in the task.

Here are the reasons why we cannot depend on members of the business community to produce valid requirements:

- They have tunnel vision. This is not a negative statement; when the process workers are doing their job, they should see the business, problem, and solution from the perspective of their own jobs and functions.
- The best person for the job of explaining the issues may not be the one given the assignment to write the requirements.
- They do not know what IT wants—they do not know what requirements are. For example, they may have no idea of what nonfunctional requirements are or how to express them.
- They may not want to commit to a specific requirement or set of requirements for fear that they will be held responsible for that commitment.
- They do not know what is available technologically.
- They may not be able to visualize the solution because they are too close to the problem and simply cannot see it.
- They are not aware of the overall implications of what they ask for and are likely to specify requirements that conflict.
- When asked for their requirements, they feel obligated to specify something, whether it is pertinent, useful, required, incidental, or nongermane.
- It comes down to this: Users and the business community do not need requirements, or even want them. They don't even want software or systems, or computers. What they want is a solution to their problem. We need the requirements so they put up with, tolerate or humor us during the requirements process so that they can get their problem solved.

Gather Information, Not Requirements

So if the users do not have requirements, what do they have? What they have is information. They provide us with information: how they do their job, what aspects of their job they want to work differently, why they perceive there is a problem, what a solution means to them, who else may be impacted by the problem or solution. They can describe the business problem, define the problem domain, identify the conditions that cause the problem, and tell us which solutions are preferable. They can relate stories, descriptions, wants, needs, gripes, facts, lies, solutions, words of wisdom, complaints, and probably a joke or two. And that is what we want: as much information as we can get. The goal of the elicitation phase of the business analyst solution cycle is to gather information, not requirements. Whoever gathers the most information, wins.

It is from skillfully elicited information that we, the business analysts, derive and define the solution to the business problem and write the solution as a set of requirements which state completely and accurately what must be done to solve the problem.

There is more to this than just changing our language and using a new catch phrase. We change our expectations when we change our language. Start by assuming the process workers really do not know what they want and make it your job to help them determine the solution to their business problem.

When workers do not have requirements, then there is no single worker you have to find to give you the requirements. You can focus on information rather than individuals. When you gather information, you focus on just getting information, all the information, as much information as you can get. The information you gather may expose other problems, shed new light on the process under investigation, clarify an issue, eliminate an assumption, or establish a good or better relationship with the responder. You determine how to apply the information. You are the business analyst.

And when you are defining requirements instead of gathering them, the product stakeholders cannot change the requirements. They cannot change scope. Only you, the business analyst, can change the requirements that you created. The product stakeholders can only change information. And changing information is never a problem.

It is during the analysis of the elicited information that the business analyst defines the functional requirements, the nonfunctional requirements, and the constraints. Analysis defines the problem and the problem domain. Analysis characterizes the solution and the impacts. Analysis may also uncover additional problems that the organization needs to address, thus increasing value.

Gathering the Information

It is a capital mistake to theorize before one has data. One begins to twist facts to suit theories, instead of theories to suit facts.
—Sir Arthur Conan Doyle, *A Scandal in Bohemia*

"How do I get the business to give us information?"

The typical approach to the elicitation process in most organizations follows this scenario:

You get the assignment to solve a problem. You meet with the problem owner, sponsor, customer or whoever wants the problem solved or is paying for it. That person gives you a briefing about the issues and suggests you talk to several people and directs you to them. You dutifully call and make appointments with the several people, or you have a meeting with them. You may prepare some questions ahead of time, perhaps mentally. You go into the meeting expecting them to tell you all about what they want, what the problems are and what they want you to do about it, so preparing a list of questions seems a waste of time and effort. You then base your entire solution on whatever those people tell you. After all, these are the people the sponsor or customer told you to see to get the answers, notwithstanding our concerns with subject matter experts expressed earlier.

The good thing about this process is that your job is relatively easy: show up and listen, and record faithfully. The additional benefit is that should things go wrong later on during solution development, you can always blame your sources, and claim to be just the messenger.

Based on what we have already discussed, you can see the flaws with this scenario:

- It assumes the stakeholders you were sent to meet have already done the work of defining the solution and you are there to understand what they have done. When you are told who to talk to, you need to figure out why you are talking with that person and what possible information they might have to help you solve the problem.
- Those who you are sent to talk to may not be subject matter experts or may have a limited view of the problem or solution. You end up being at the mercy of the problem owner, sponsor, and subsequently those you are sent to see. In other words, they are deciding what information you need to prepare your solution.

- You end up with only one solution—theirs. Your job is to define and assess as many solutions as you can. In this way you can better assure the organization that the best solution has been chosen.
- There is a marked tendency for those who have been assigned by an upper-level manager to give you information to instead give you solutions because they naturally perceive that is their role.
- The information you get may not be complete. It is complete as far as they know, but they could be leaving out large chunks of valuable information. Since upper-level managers will usually refer you to mid-level managers rather than the users, you may be talking to people who are years removed from the firing line or the user interface.
- The information you acquire is going to be haphazard and disorganized based on the whims of those you have been told to check with. You want to gather the information in an order that makes sense to you and makes your job of defining the solution easier.

In the end, you are gathering information for yourself, to help you define or verify the solution. Only you know what information you need to do your job. You should control the sources of the information, what information you acquire, and the order in which the information is collected. What I recommend is this: Instead of first identifying who to talk to, determine the information you need to define and solve the problem and then determine what is the source of that information. Use an information-gathering plan to structure your investigation.

Information-Gathering Plan

"How do I structure the elicitation phase so that I can get the maximum good information?"

Gathering information calls for an information-gathering plan (IGP). Ask any reporter or author of a nonfiction book. They prepare a plan to get the information before they start investigating or asking questions. Why? To save time by giving you:

- A clearer idea or vision of the information you need to acquire to define or solve the problem.
- Fewer information gathering sessions.
- More usable information out of each session.
- Fewer people to gather information from.
- A more efficient order of acquisition.

I know what you are thinking. Another document? Another formal step in the process that already takes too long! The information-gathering plan is not a formal document. It is not intended to be. It is a temporary document, simply a way of organizing your information acquisition process. The IGP can be written on a flip chart, a whiteboard, or a piece of scrap paper. It is more effective to use a whiteboard or central display if you are working with a team and are splitting the information-gathering duties among several team members.

The IGP consists of four parts, created in the following order:

1. What information do you need to understand the problem or the problem domain?
2. Where are you going to get that information? Where is it most likely located or who might have it?
3. How are you going to acquire the information?
4. In what order are you going to collect the information?

The IGP can be created in a matter of minutes for small projects or work requests. For larger efforts, it should be a specific exercise with some time allocated to its completion, and typically involves a group of business analysts. Even in the largest of efforts, creating the initial IGP takes less than an hour.

What Information?

First you gather enough information to ensure you can define the product scope. Then get the information to understand the conditions and circumstances that have caused the problem: the problem domain. Acquire information to solve the problem in terms of risk and impact to other systems and processes, and to produce the solution document.

Functional Information   First define the problem domain (See Chapter 12). What is/are the business process(es) that contain the problem? Who is involved? What do they do? How do they do it? In general, obtain:

- Information about actual events to help you understand the conditions causing the problem.
- Information about concepts the users think about when performing a task.
- Simulations of tasks so that you can observe the process worker's thinking, responses, and actions.
- Work-arounds that have been created to bypass process or system limitations or defects.

- Stories and anecdotes about the process.
- Complaints and issues with the process.
- Identification of the individual strategies that are used to perform tasks and activities under specific, especially abnormal, conditions.
- Identification of the triggers that are used to indicate when to apply the strategies.
- The flow of information through the entire business process.

Supporting Information   While you should be focusing on the functionality of the process and/or system, you may hear the responders talk about other concerns, such as appearance of the user interface, speed of operation, reliability of information, and so forth. In distributed environments or Web-based systems there are other concerns that the process workers and users may not be privy to or have an interest in, such as security, location of data, update synchronization, reliability, maintainability, and so forth. All of these nonfunctional aspects of the system and process affect the overall quality of the users' experience and the success of the solution.

How to do it: Brainstorm the information you need to acquire. You may do this alone, although getting a couple of other business analysts together for fifteen minutes to brainstorm what information each participant would want to know about generates more questions and topics. The other business analysts do not have to know anything about your project or problem. Every question or information topic that comes up is a possible candidate to be included. Put every question or subject on the whiteboard.

Then ask these questions of all the topics on the list:

- Why will this information help?
- What will I learn from a valid answer to this question?

These questions are filters to help eliminate specious questions that will not provide much information. This saves time and keeps the process on track.

Table 11.1 is an example of an information-gathering plan partially filled out for the accounts payable problem defined in Chapter 8.

TABLE 11.1 An Information-Gathering Plan

What Information	Source	Method	Sequence
The layout of the current vendor tables.			
What does the voucher entry process look like?			
How do they do vendor entry?			
What is the data that goes into computing the payment terms for vendors?			

Sources of Information

"I didn't get good requirements because I talked to the wrong people."

Once you have established what information you need, identify the source for the information. Where do I go to get the answers to my questions and obtain the information I need? The main sources are hard copy and process workers.

Hard Copy   First consider hard copy sources, if any, such as system-generated reports, current existing documentation, help desk logs, defect or problem tracking report logs, intracompany correspondence on the subject, pertinent government regulations, corporate policies and procedures, the corporate annual report, the organization's Web site, and so forth. In the BABOK, this method is called document analysis.[2]

When defining the problem domain, hard copy information is preferable over human information because it has less bias. Many people usually contribute to publicly available, persistent documentation—the author, editors, technical writers, legal department, HR, managers, and others—all of whom will make sure the information is accurate and objective. When talking with an individual we get that person's opinion of what things mean based on that person's experience in the subject area. The better approach is to review the written word first and then get an individual's opinion of what it means. That way your analysis and solution are not biased.

Process Workers   A common complaint among business analysts is, "The business analysts don't get to meet with the lower-level people in the organization who actually use the system." Some managers feel that it is their duty to be involved with everything down to the minutest details. And some feel as though their minions are not capable of defining what they need in a system or new process. And there are those managers who do not want to take their workers off the production line to talk to IT or a business analyst, and feel compelled to represent their workers and provide the information. In all these cases, the business analyst does not get the benefit of talking directly to the people who are going to use the solution that is defined.

Whenever possible, though, the best information comes directly from the end users. When you cannot identify specific product stakeholders who have the information you seek, add the item to your information-gathering plan list—"Who can provide me the information about XXX?"—and then decide who you need to ask to answer that question. Usually the point of contact or problem owner can direct you to the source to the information you need.

"How do we deal with people who are against [name of project]? Especially after it's already installed?"

Hostile Stakeholders

One thing we know is true: No matter how good the change is for the organization, no matter how beneficial, no matter how universally acceptable, there will be some constituent somewhere who is against the change, real or perceived; someone who believes that he or she will be disadvantaged by the change. And, yes, we have to consider those who are against the changes we are making as stakeholders, although it is sorely tempting to ignore those who are against the changes because they have nothing positive to offer us in our pursuit of the solution.

It is fairly easy to determine who may be against an initiative that replaces workers with automation. It is not as easy to identify the hostile stakeholders when everyone in the organization appears to benefit from the change or the change is clearly a positive move for the organization. It is even more difficult when upper-level management dictates the change.

🔍 Example

A large New England insurance company had a department of 20 or so accountants who manually computed the financial penalties of taking early withdrawal of various investment instruments, such as 401K plans and IRAs. The company decided to automate that process but needed the expertise of the people in the department, the people who would be losing their jobs when the new system was installed. The company put the project on a fast track to be completed in an aggressive timeframe dependent on getting the specifications from the accountants. Management realized that gaining full cooperation out of the people in the department would be difficult once they were aware that their jobs would be eliminated, and even if management got cooperation, it certainly would not be quickly forthcoming as workers would stretch the project as long as possible or until they secured other employment, at which point they would take the information with them. Management got together with the business analysts and HR and determined where to place every one of the accountants, some with lateral positions and some with promotions. Then management and the business analysts announced in a meeting what was happening, being honest about the loss of the current positions. Once the workers discovered where they were going when the new system was implemented they eagerly participated in the information gathering and collaborated fully with the development of the new system. The system was implemented forty days ahead of schedule and only one of the workers left the company. He chose retirement.

Process workers show their reluctance to embrace a new change in a number of ways: conflicting requirements that have to be resolved before the system is implemented, late changes when new information arises that could have been presented earlier, misinformation, missed meetings, delays of all sorts, marked lack of interest, and so forth.

As you discover the hostile stakeholders, you will have a natural tendency to avoid them. Who needs the confrontation? They are not going to add anything positive anyway. Let management handle them after the change is implemented. Suppress the tendency and seek out the hostile stakeholders and request time with them to listen to their reasons why the change should not be implemented.

Why should we do that?

Because those in favor of the change, those who will be advantaged by it, will only tell us what good it will do. They will not describe the risks and downsides of the project. They will be afraid that such mention will cause the project to be delayed or even canceled if the risks or downsides are too great.

Those against the change will present all the bad things that could happen, both real and imagined in an effort to scuttle the project. You want to hear those bad things as early as possible in the solution life cycle (SLC) so you can address them and evaluate them. You also want to adjust your solution to reduce or eliminate those adverse implications of the resulting solution.

✔ Tip

When meeting with hostile stakeholders, the purpose is not to convince them that their objections are groundless or mollify them into being in favor of the changes. Go into the meeting with maximum empathy. Try to understand what their opposition to the change is all about. When the hostile stakeholder presents an objection, don't immediately point out that the objection is wrong. For example the stakeholder repeats rumors you know to be untrue that his team will be laid off when the system is implemented. Your first impulse is to disabuse him of the notion to relieve some of his hostility. Do not agree, simply listen and empathize. Should you start countering all his objections, he will assume that is the only reason you are there and stop giving you objections, problems, and obstacles. Your goal is to get as many objections to the change as the hostile stakeholder can give you. Later, you can revisit the stakeholders to tell them how their objections were addressed, assuming the objections were addressed. The neat thing about this guerilla tactic is that the stakeholder will naturally assume you had something to do with the removal of the obstacle (for example, telling him that there will be no layoffs) and you have gained an ally.

TABLE 11.2 Adding the Information Source to the IGP

What Information	Source	Method	Sequence
The layout of the current vendor tables.	Data Dictionary Database Administration		
What does the voucher entry process look like?	Accounts Payable Policies and procedures manual Voucher entry team		
How do they do vendor entry?	Vendor entry clerk		
What is the data that goes into computing the payment terms for vendors?	Accounts Payable manager Purchasing manager		

In your information gathering plan, fill in the second column identifying at least one source for each question or topic about which you need information as shown in Table 11.2. Identifying multiple sources as shown in the table is a good idea. It identifies backup sources in case the initial source does not have the information or is not available, and it gives you the confirmation source. Note that the source is listed as a category, such as database administration, or a role or position, such as accounts payable manager, even when you know the name of the person in that role. Additional information may come up that may indicate a new person as a better source.

Methods

The method selected for information acquisition must adhere to the policies and culture of the organization. While prototyping the user interface may seem to be the ideal way of collecting information on what the users want to see in front of them while they work, the organization may not be able to withstand the perturbation of frequent prototyping sessions that take users off the production line. Other methods of gathering information may be frowned upon by culture or policy. I was at one organization where individual interviews were simply not allowed. Policy stated that all information had to be gathered in a meeting format. I was at another organization where the culture of the organization prohibited the use of joint application development (JAD) sessions.

Example information-gathering methods as defined in the BABOK are:

- Interviews.
- Meetings or focus groups.
- Document analysis.
- Observation.
- Prototyping sessions, including storyboarding.
- Interface analysis.
- Questionnaires and surveys,
- Brainstorming.[3]

In addition to the methods listed by the BABOK here are some other methods:

- Use case sessions.
- Workflow analysis.
- Technology demonstrations.
- Business case analysis.[4]

Some methods are for gathering new information, and others are better used for follow-up. For example, phone and e-mail are good follow-up sources of information provided a face-to-face meeting has taken place first.

Table 11.3 shows when to use the various methods. The four categories (clear, stable; clear, unstable; unclear, stable; and unclear, unstable) apply to both the business community and the business analyst. What may be obvious and stable to the users may become cloudy and changing to the business analyst as he gathers more information.

TABLE 11.3 When to Use Methods of Elicitation

	Clarity of Requirements	
	Clear	Unclear
Stable requirements	Interview	Meetings
	Surveys	Interview
	Phone surveys and phone interview	JAD sessions
Changing requirements	Brainstorming	Dynamic prototyping
	Observation	Observation
	Static prototyping	

TABLE 11.4 Methods Added to the IGP

What Information	Source	Method	Sequence
The layout of the current vendor tables.	Data Dictionary	Read	
	Database Administrator	Interview	
What does the voucher entry process look like?	Accounts Payable	Read	
	Policies and procedures manual	Observation and Interview or meeting	
	Charley and/or member of voucher entry team		
How do they do vendor entry?	Vendor entry clerk	Observation and Interview	
What is the data that goes into computing the payment terms for vendors?	Accounts Payable manager	Interview	
	Purchasing manager	Interview	

On your information gathering plan, fill in the methods column with at least one method for each source, as shown in Table 11.4.

Sequence of Acquisition

You could arrange to conduct the interviews in alphabetical order by last name of the responder. Or you could ask when people are available and try to fit everyone into your schedule based on what time they request. A better approach is to acquire the information in a logical sequence. Using the IGP, you can determine that logical sequence.

The sequence of acquisition refers to the order in which the information is gathered, but keep these guidelines in mind:

- A top-down approach, starting with the highest level in the organization with an interest in the problem, provides a better overall sequence, and it allows you to obtain downward references.
- You will need to know some information before you know enough to determine what else you need to know, which typically means working from big picture down to details.
- There may be a pecking order, or political structure, dictating who must be talked to first and last.

When filling in this column on the IGP, do not allocate time slots or make appointments at this point. All you want to do at this point is structure

the flow of information into a logical sequence. You need to decide whether you will observe the workflow or conduct interviews first. When interviewing, you need to determine the order in which you will interview people.

The hierarchical imperative is what kicks in when management overrides the natural sequence for acquiring the information or otherwise dictates information gathering. The most common example of this is when you plan on meeting with the senior person in the business organization first, to get a broad picture of the problem domain and any business constraints or critical success factors that might come into play. The senior person says that they would be happy to meet with you for a half hour, after you have talked to everyone else. You quickly have to rearrange your information gathering plan to get the big picture information elsewhere, and conduct your executive interview last. That is hierarchical imperative.

🔍 Example

Hierarchical Imperative

I had asked the manager of a large effort for time at the beginning of my investigation for an interview. The manager told me I had six minutes at the start of the project for a quick briefing (and he gave me exactly six minutes of his time) and stipulated that I could have a full interview, but only after I had talked to everyone else and analyzed the information. When we got together for the end-of-elicitation meeting, I was expecting to get a picture of the overall operation of the work and perhaps confirmation of some of my observations about the processes. What he was looking for was a report on what I found out about his people and what they were doing. That meeting was quite an awkward situation.

Sequence your information-gathering plan as shown in Table 11.5. Note that the sequence relates to the information to be gathered and not the source.

The information-gathering plan is a roadmap for gathering information. Just like any other plan, it is subject to change. Throughout the investigation, and even during the analysis and documentation, new questions will arise, as will the need for new information to clarify issues, resolve ambiguities, get more details, and so on. Just as you would change your planned trip through Kansas to the West Coast when you see that the largest ball of twine in the world is just an hour away, so you keep adding the new questions and information topics to

TABLE 11.5 Information Gathering Plan Sequenced

What Information	Source	Method	Sequence
The layout of the current vendor tables.	Data Dictionary Database Administrator	Read Interview	2
What does the voucher entry process look like?	Accounts Payable Policies and procedures manual Charley and/or member of voucher entry team	Read Observation and Interview or meeting	4
How do they do vendor entry?	Vendor entry clerk	Observation and Interview	3
What is the data that goes into computing the payment terms for vendors?	Accounts Payable manager Purchasing manager	Interview	1

your information-gathering plan road map, with associated sources and methods. As you acquire the information you cross the item off the information-gathering plan. When do you have enough information? When there are no items left on your information-gathering plan.

✅ **Tip**

We place our information-gathering plan on a whiteboard, at least the list of the information we need to gather. As we gather the information we cross the item off the list. We don't erase it. When we require more information as a result of analysis we add the topics to the list on the board. Sometimes the list flows into flip chart pages stuck to the wall. We may annotate the item to point to where the information is located in our files. This simple low-tech approach yields two benefits. In longer problem-solving efforts we may forget we have already obtained certain information and this list keeps track of the information we have acquired. Also there will come a time when topics are being crossed out faster than they are being added and we can see real morale-building progress. Eventually when all the topics have been crossed off, indicating we need no new information to solve the problem, we can visually see that the job is done.

Information-Gathering Session

With the information-gathering plan completed you can now start gathering the information needed to produce a solution through a series of information gathering sessions. An information-gathering session is any form of interaction in which information is transferred from source or sources to the business analyst, and includes any and all of the methods identified earlier. The form of the information may be in notes taken by the business analyst, diagrams constructed in collaboration with the source(s), or documents provided by the source(s).

Regardless of the method of information gathering, there are five stages in a successful information gathering session:

1. Preparation
2. Introduction
3. Body—questions and answers
4. Close
5. Follow-up

Let's take a look at how each of these stages helps us better gather information.

Preparation

Preparation for an information-gathering session sets the stage for success. Preparation does not take a long time; usually about half the time allocated for the session. Preparation activities are important to ensure you get maximum value for the time you and the responders spend in the session.

Earlier in our description of the traditional elicitation scenario, I described the business analyst arriving at the information-gathering session with only one question in mind: "What are your requirements?" Because you are no longer assuming that anyone has requirements, and because you want control of the information you receive, you need to prepare the questions you will ask to get that information.

Write down the questions you intend to ask. Questions taken as a whole when answered should provide information to achieve your objectives. Why bother to write them down? Here are some reasons given by business analysts for writing down questions:

- Remember your questions.
- Keep track of the questions already asked.
- Share questions among multiple interviewers.
- Validate question content.

- Verify good phraseology.
- Document for the future what you asked just in case.
- Document source of information (traceability).
- Provide consistency in the questions you ask.
- Provide a way of organizing the questions.
- Demonstrate that you are prepared.
- Stay on track—you can remember what the next question was when the responder goes on a tangent.

Objective   You need to know what your objective is for each information gathering session. Whether an interview or a meeting, you should know why you are taking people's time. Jotting the objective down before you write the questions allows you to organize the questions around the objective. The objective is typically an entry in the What Information column of the information-gathering plan.

The objectives must be specific and achievable. An objective to understand the current process begs the issue: Can you fully understand the current process as the result of an interview? The objective to learn more is not specific enough, and too easily achievable. An example objective would be to determine the population of the organization currently affected by this problem.

> *What we think of as the moment of discovery is really the discovery of the right question.*
>
> —Jonas Salk (1914–1995)

Asking the Right Questions   Another common comment from business analysts is, "We don't get good requirements because we don't know the right questions to ask."

> "But how do I know what questions to ask the users?"

The real answer to knowing the right questions is simple: Keep asking. When you ask enough questions, in and among all the questions you ask are the right ones. As long as you listen well and keep the focus on the problem or the solution, it does not matter which questions are right. In the end, the right questions are those that get you relevant information.

You learn through experience as an investigator what questions to ask and how to ask them. You also learn that each person from whom you gather information needs a different approach to get the same information. Asking the same questions of everyone will not necessarily get you the information that you need.

Preparing the Questions   Whether you are interviewing, observing, or con-
ducting a meeting, you are asking questions. The purpose of information
gathering sessions is to gather information, and the primary way of gather-
ing information is by asking questions. Determine the questions you ask and
the information you want during the preparation phase.

The type of question asked is based on the type of information desired
and the type of person you are interviewing. For example, you ask different
types of questions when interviewing an upper-level manager than when
you have a meeting of several process workers. Different questions are used
to determine how the solution fits in with the overall strategy of the depart-
ment rather than to elicit details about the use of a particular element of a
system. The following are some of the more common types of questions
you might use during an information gathering session:

- **Closed-Ended.** Closed-ended questions require a short answer from a
  finite list of possibilities. "Yes" and "No" are the most common close-
  ended responses. Other closed-ended questions are about dates, lon-
  gevity, and asking for explanations of why the user made a specific
  choice when describing the process. Closed-ended questions are used
  to get quantitative answers. For example, you might ask an initial ques-
  tion of a responder during an interview, "How long have you worked at
  this job?" This is a closed-ended question answered with a number of
  months or years.
- **Open-Ended.** Open-ended questions are those that are asked in such a
  way that the responder has the option of answering them any way the
  responder feels: a soliloquy, a shrug, or a statement. For example, you
  might ask the same responder, "How do you like this job?" which might
  evoke a smile, a monosyllabic response, or a dissertation. This is an
  open-ended question.
- **Context free.** Context-free questions are questions that can be asked in
  any situation and do not relate to any particular project. They tend to be
  questions about the big picture. These questions can be asked about
  any project at any time. There are 13 context-free questions listed in Ap-
  pendix E that, when answered completely, give you a wealth of infor-
  mation about the problem.
- **Meta.** Meta-questions are questions about questions such as "Can I
  ask about . . . ?" or "Is this a good time to talk about . . . ?" Use
  meta-questions to keep the session from becoming too much like a
  legal deposition
- **Contextual.** These questions are specific to the problem at hand or the
  solution being considered. They typically could not be asked in any
  other context. A contextual question might be: "Tell me, Charley, what
  time does happy hour usually start?"

- **Leading questions.** Leading questions are asked primarily for confirmation. They are a form of closed-ended questions—typically beginning with don't, can't, and isn't—for which the answers are expected to be yes or no to confirm or contradict a previous statement.
- **Validating questions.** Once you have a solid list of questions, validate that list with the following:
  - If I get good answers to all these questions, will the objective(s) be achieved?
  - Are any questions misleading?
  - Might any question go unanswered or be avoided by the responder?
  - Will any question cause the responder to ask for clarification?
  - Does the responder know enough to answer the questions?
  - Are any throwaway questions (if they are not asked and answered, no good information will be lost)?
  - Do any questions get the same information as other questions?
  - Do I really want to hear the answer to that question?

When the questions have passed muster, you are ready to start using them.

''We don't get good requirements because of suspicions on the part of users about why we are asking questions.''
''The businesspeople don't know what to tell us.''

Introduction

The introduction sets the stage for the rest of the session, so it is important to get off on the right foot. Instead of starting out by stating the objective you wish to achieve with the session, as in, ''Hello, Charley. I'm Steve. I am here to understand how you enter the vouchers,'' start the session by expressing the following (captured in Chapter 9):

- This is the problem we are here to solve.
- This is the vision that we see occurring as the result of solving the problem.
- This is why it is important to the responder(s) personally.

The opening statement then goes something like this: ''Hello, Charley. I'm Steve. I want to talk to you about the time it takes to do voucher entry. I understand that it is just taking too long with all these vendor payment terms that have to be entered. We want to have a situation where all of the terms are automatically entered like the other data, and all the vouchers are completed in one day and you get to go to happy hour on time.'' The advantage of this approach is:

- It focuses the session on the problem and limits excursions and sidetracks.
- The responders or reviewers tend to be more motivated to provide information that solves the problem.
- You, as interviewer or moderator, have a point to refer to whenever there is a question or potential conflict.
- By observing the responder's reaction, you can get a good feeling whether the responder has the information you are seeking. When the responder nods in agreement with the problem, acknowledging it, you know he has the information.

Body

The body of the information-gathering session is all about questions and answers. Here is where the business analyst's soft skills come into play: active listening, empathy, body language (yours and theirs), selection of questions to ask, sequence of questions, influencing skills, and so forth. The goal is to gather as much information as possible in the given timeframe.

Another concern for business analysts is, "*We didn't get good requirements because the requirements person had too much knowledge about the problem domain and didn't really find out what the users really needed.*" You must remember that:

- Responders do not give you information they believe you already know.
- Responders stop giving you information when you have corrected them or challenged the information they have given.

One key ingredient in the success of the information gathering is Miller's Law. George Nathan Miller created this law to achieve successful communication:

> To understand what another person is saying, you must assume that it is true and then try to find out what it could be true of.

John Winter, Vice President of Product Development at the International Institute for Learning, suggests another way of looking at this: you should "listen naively." Regardless of how much time you have spent investigating the situation, approach each new information-gathering session as though you are a blank slate and the responder is going to fill it; in other words, assume every responder is an SME. When you pay attention and approach the session knowing nothing, there is never an information gathering session in which you do not learn something new, gain some new information. When your plate is empty, they will fill it with information.

"We don't get good requirements because we don't ask enough questions."

✔ Tip

For the most part everything that comes out of the business analyst's mouth during the body of an information-gathering session should be questions. Even comments on questions or responses should be in the form of questions. Here are some examples of questions that will generally increase the flow of information:

- I am not sure I quite understand that last point, could you clarify for me?
- Do I understand that you are saying . . . ?
- Could you elaborate on that for me, please . . . ?
- If I were a voucher enterer, how would I know . . . ?
- Can you describe that in a different way?
- Can you explain that to me as though I am six years old?

Think about each question before you ask it so that the question is not redundant. Phrase the questions so they do not invite tangents and sidetracks. In a group setting, when there appears to be some contention about an answer, ask more questions for clarification, or simply ask the responder to explain the answer more fully. Many times a simple misunderstanding causes a conflict that you can clear up beforehand so that the contention does not surface to derail the session. Set a goal for any information-gathering session that everything you say during the body of the information-gathering session is stated in the form of a question.

Close

When planning the information-gathering session, you specifically allocate a percentage of time (usually about five minutes in an hour-long session) for the close. You want to leave the information-gathering session with a firm feeling that you can come back again, if necessary, to ask more questions. You want to gracefully close the session so the responders feel a goal was achieved, the information provided was valuable, and their time was not wasted.

There are three questions to ask before ending the session:

1. Is there anything I failed to ask about?
2. Is there anything you would like to ask me?
3. Do you know of anyone else who might have information about this?

Between the body and the close there is a distinct break. You are announcing to the responders that the question/answer session is over and the interview or meeting is coming to an end. This break can be as simple as saying "We only have a few more minutes in our allotted time, so I would like to close with a couple more questions."

Summarizing   In theory, summarization shows you have been listening because you can paraphrase to the responder what was said, and it gives the participants a chance to voice objections or add new information.

In practice, you do not want to summarize at the close of an information-gathering session. Consider the purpose of a summary. You summarize so that the responder can confirm the information you received. You only have a few minutes left in the interview, and ending on time is an absolute rule of the information-gathering session, regardless who the responder is. When there is any discussion about the content of the information and you summarize for the responder, you are sure to go over time. The responder will not remember what caused the interview to run over, only that it ran over. Instead, handle the summary as part of the follow-up (next section) and use the graceful exit: "Thank you for your time and the information you provided. It will go a long way to helping us solve the problem." For example, "Thanks, Charley, I now have a good understanding of how voucher entry works. Your suggestions will help us speed up the process and probably get you back to happy hour on time."

Follow-Up   The follow-up occurs after the information-gathering session and consists of summarizing the information received and sending the summary back to the responder in an e-mail or other communication along with a thank you for their participation. You do so after analyzing the information. Regardless of how long it actually takes to review and analyze the information, wait at least an hour or so before sending the summary. That shows that you have taken the time to review the information received and increases the responder's impression that the information they contributed was valuable. This goes a long way to getting more information from that responder in the future.

This technique has several advantages over the closing summary:

- It allows you to review the material and ask any other questions as a result of your first cut analysis.
- It tells the responder that the information they provided was valuable, which increases the responder's interest and commitment to solving the problem.
- The responder may have some additional information that he thought of after the session and this e-mail affords him the opportunity to express that information.
- The responder may have second thoughts that occur after the session and correct or augment the summarized information.

While it does not happen all the time, when you do get a response back to your e-mail summarization, it is like getting a second interview free; that is, you did not invest any more time in the acquisition of the information than to read it.

Follow-up information-gathering sessions come in two flavors: following up with the same person to add more information or additional clarification, or following up with another person to confirm the information of a previous session. Follow-up sessions with the same person should always be shorter than the previous session.

The information from any given session may shed light on or change the interviewer's opinion, idea, or concept. Not only will subsequent information cast doubts on the veracity and accuracy of previous information, but it also challenges the business analyst's attitude about the information gathered to date. Use analysis and additional confirmatory sessions to reconcile conflicting information. Confirmation of information gathered and of your interpretation of that information is important as a checkpoint when subsequent information yields contradictory facts. Be careful of falling into the trap of only hearing and eliciting comments from responders that confirm the views expressed in previous interviews, especially if they coincide with your preconceived notions or the conclusions you are building based on earlier information-gathering sessions.

Second Source   As any investigator or newspaper reporter knows, regardless of the presumed veracity of the source, any statement elicited from a source must be confirmed or corroborated by a second source. This is another form of confirmation.

The successful business analyst comes to no conclusions and bases no requirements on a single source. When a process worker describes how a process is performed, the business analyst confirms that description by observing it or asking another process worker to describe the same process.

Solving Common Information-Gathering Issues

Information-gathering sessions have a number of common issues regardless of the type of session (interview, meeting, etc.). These are ones often expressed by business analysts (though they are not by any means all of the issues).

They Do Not Know What They Want

Business analysts sometimes get frustrated because it is hard to get the users and process workers to identify what will solve their problem. The reason they have trouble knowing what they want is because they do not really know what they have and what is available to them.

What to do: Try offering different solution scenarios that the responders can reject, modify, or replace with an alternative. Once responders start work on a solution, they start realizing what they really want.

Example

How hard would it be to . . . ?

There are several reasons for the impression that users do not have a clue. One of the common phrases I hear during late proto-typing or even after the solution is delivered, is "How hard would it be to . . . ?"

The first times I heard this phrase, my reaction was not kind. I may not have actually said, "If you wanted that, why didn't you ask for it in the beginning?" but I am sure I thought it loudly, and after-ward, walking down the hall, I am sure my fellow programmer and I groused about stupid users never knowing what they want, and so on. But later I realized that what was stated was the second half of a sentence, the first half of which went unsaid: "*If you can do this,* how hard would it be . . . ?" In other words, they had no idea of the technological possibilities available to them until they saw it in the prototype. And these are the people who are supposed to be telling us exactly what they want so we can give them the solution! It is like asking someone what they want on their dashboard when they are not aware that the automobile has been invented. Only when they see the dashboard and how it is used will they come up with what they want the dashboard to look like.

They Can't Communicate What They Want

One of the unique aspects of the information gathering session with the business community is the challenge of divining what they want when many times they are not sure how to express it. Sometimes their reticence is due to a belief that they have to express their desires in technical terms to be understood, and they do not know the words.

What to do: Assume either the translator or educator role to assure them that they can describe their ideas and solutions in their own terms and you will translate them where necessary.

When you are just asking for information from the responders they find it easier to express the answers because there is no imprimatur of responsibility. You can use your elicitation and investigation abilities to draw out what they are really trying to say, playing the facilitator role. As long as the responders do not feel they will be held accountable for a solution they voice, they will provide information and potential solutions.

"We don't get good requirements because the business doesn't want to do requirements."

They Don't Want to Do Requirements

When faced with this obstacle, the easiest path is to not do requirements. Instead, ask them for help in understanding the problem domain. Focus on the holistic view of defining a solution for their problem. The process workers usually are not against spending time with you to solve their problem, they are against the documented requirements that they feel they have to produce. Remember that they don't want requirements they want solutions.

They Want Everything, Just in Case

"They never want to stop meeting and telling us what else they want. Even with stuff that they aren't sure of they say to put it in the system anyway. The project sponsor says 'Better safe than sorry,' but we can't do everything and still be on time, and my project manager is getting upset."

There are two reasons the business community starts demanding everything they can think of. First, they do not know what the real problem is so they ask for everything on the chance that something they ask for will solve the problem. Second, they do not know when they will get another chance to change things so they want to get all they can while you are cataloging their requests.

What to do: Resolve the first issue by defining the real problem so they can focus on just what they need to solve it. The second issue can be resolved by establishing an incremental delivery of the functional goals. Once the business community realizes that each problem solved is another step in the overall business process improvement effort, they do not worry about not getting everything the first time.

Hidden Agendas

''We can't get good requirements at [name of organizational entity]. There are too many hidden agendas.''

''The requirements we got were wrong because they were influenced by personal objectives.''

''How do you deal with managers who come to meetings with hidden agendas?''

Participants often come to information-gathering sessions with hidden agendas, such as a personal bias for or against something. They answer questions in a way to further that agenda, which means the answers could be purposefully misleading, incomplete, or completely false. It is difficult to tell when there are ulterior motives behind responses or actions.

What to do: Observation is the best way of telling that a hidden agenda is in play. When you observe changes in body language or vocal characteristics in response to a question or new topic, usually something else is going on. It is probably not a good idea to voice your suspicion or challenge the responder. Simply knowing that there is some information that is undisclosed may be enough to temper your analysis of the information. Knowing a potential hidden agenda exists may increase your investigatory efforts.

Iterative Information Gathering

Information is gathered iteratively. You simply are not going to get all the information to answer all your questions in one sitting. Plan on it. The same individuals attend multiple sessions. Each session is separated by analysis performed on previously obtained information where you raise more questions that have to be answered. The subsequent information clarifies, elaborates, or refutes the information gathered earlier. And each new time you engage the stakeholders they may have new information, or provide a different view from the previous session, based on recent events or reconsideration.

The two primary methods for information gathering in business today are the interview and the meeting. Let's apply the stages of the information-

gathering session to these methods and address some of the specific issues business analysts face.

Interviewing

The best approach is to quietly and persistently probe for information, not like a grand inquisitor but rather as a humble human being seeking genuine advice.

—Herb Cohen

While it appears to be going out of style in an agile world, the information interview practice is still the best means of gathering information we have available to us. Though it may appear easy on the surface, the successful information interview is not an easy task to accomplish. It is hard work. Then why choose the interview over other methods of acquiring information? Because it allows you to:

- Obtain more relevant information.
- Get the information directly, therefore, faster than other methods.
- Get the human spin on technical processes.
- Acquire background information.
- Follow up immediately to reduce misunderstanding.
- Learn prejudices and feelings about the process or system.

Types of Interviews

There are two types of information interviews: the structured and the unstructured. Each has a different purpose. The structured interview provides more focus and obtains specific information, while the unstructured is more general and sometimes gathers more information, especially information on which the interviewer had not expected. Usually interviews are combinations of the two approaches.

Structured   In the structured interview you ask a set of predetermined questions and get answers. The questions are like a script that must be followed word-for-word. There is little variance to the script.

The interviewer needs skill in asking questions so answers are not biased because of the way the questions are asked. A responder tends to try to give the interviewer the answers the responder thinks the interviewer is looking for, and bias on the part of the interviewer causes the responder to color or shade the answers, leaving out details or slanting them to suit the interviewer's bias. Reactions to an answer that show the interviewer's bias will tend to stem the flow of information, especially if the responder has difficulty with the interviewer's reaction or misunderstands it.

The structured interview is sometimes used for subsequent or follow-up interviews when the business analyst has only a few questions, or for getting specific information, such as how a process worker performs an activity.

Unstructured   In the second type of interview, you also prepare the questions in advance. After you ask the first question of the interview, it proceeds on its own. You ask the next question based on the response to the previous question. This requires more skill on the part of the interviewer to keep the discussion on track, to curtail segues without suppressing the flow of information, and to make sure that your information objectives are achieved.

Conducting the Interview

Once you have decided to conduct an interview and determined which type of interview to conduct, you go through the five stages of the information-gathering session.

Interview Preparation   Prepare for the interview the same as you prepare for any information-gathering session. In the initial interview with a product stakeholder it helps to know something about the person you are interviewing. This knowledge gives you the ability to structure the interview more appropriately and to select and organize the questions to ask.[5]

Interview Introduction   In the initial interview, it is always possible the person sitting opposite you in the interview situation does not have the information you seek. During the introduction, qualify the responder by asking some general questions that give you a sense of the responder's experience, capabilities, and knowledge. At the same time, these questions help put the responder at ease because the questions are usually nonthreatening, and that helps establish the rapport. Typical questions asked during the introduction involve length of time the responder has been with the organization, what he does in his current position, his work with the system or process, and so on.

Rapport is important in an interview, more so than in other forms of information gathering. I am not talking about creating a long-lasting relationship. I am talking about the rhythm of the interview, the questions and answers. You base your solution primarily on the information you are gathering so you do not want the responder censuring his answers, embroidering, slanting, or spinning the stories, or in any way altering the facts and/or his opinion of the facts, inadvertently or on purpose.

Rapport also means that the responder trusts that you will not ask any questions that might cause embarrassment or unease, and that the answers

given will be received without bias or negative reaction. This is called a climate of confidence.

Interview Body During the interview body your goal is to obtain as much information as possible. This is the question and answer section of the information-gathering session. The object is to gather information. The more information, the better. And the way you get information is to ask questions. The more questions, the better. Your goal should be to conduct the entire body of the interview without making a declarative statement. Only questions should emanate from your lips.

The information you receive will not only come from verbal responses. You also gain information from:

- The words used to answer the questions.
- The way the answer is phrased.
- The pause before the answer.
- The pauses between words, phrases, and thoughts.
- Facial actions and reactions.
- The body language and changes in the body language.
- Side comments, tangents, and segues and the trigger that caused them.

Table 11.6 shows a partial list of what your fellow business analysts consider activities that contribute to successful acquisition of information during

TABLE 11.6 Interviewing Guidelines

Contributors to an Effective Interview	Behaviors/Activities to Avoid during Interviews
Show empathy.	Chewing tobacco.
Be prepared for negative comments.	Repeating yourself.
Keep focus.	Go off on tangents.
Speak understandably and clearly.	Criticizing.
Act interested in what they have to say.	Referring to others or to hearsay from others.
Admit lack of knowledge (rather than faking it).	Using jargon (yours or theirs incorrectly).
Pause appropriately—don't be motor mouth.	Taking sides on any issue.
Have a few extra questions (beyond prepared ones) in case there is extra time.	Stressing negative aspects of existing systems/processes.
Watch body language.	Interrupting.
Be sensitive.	Asking age.

(*continued*)

TABLE 11.6 (*Continued*)

Contributors to an Effective Interview	Behaviors/Activities to Avoid during Interviews
Give signs of understanding.	Monopolizing the interview.
Smile.	Chewing gum, eating food, etc.
Focus the questions on the objectives.	Answering cell phone.
Ask questions that flatter interviewee by allowing him to demonstrate his knowledge of subject.	Talking about sex, religion, or politics (during introduction, for example).
Control the interview.	Being confrontational.
Stay calm and positive.	Being obsequious.
Pay attention.	Being arrogant.
Keep energy level even.	Putting the interviewee on the defensive.
Classify answers.	Dominating the other person.
Get managers away from their offices.	Asking leading questions.
Give them a chance to answer.	Going to the interview hungry.
Paraphrase longer answers to show you listened and understood.	Asking extraneous or unrelated questions.
Look at them occasionally.	Rolling eyes or other demonstration.
Keep asking questions.	Putting people on the spot.

an interview, and also those activities or behaviors you should avoid because they tend to stem the flow of information.

Interview Close   Leave enough time for a graceful close. Do not summarize. Thank the responder for the time and information.

Three questions to ask at the close are:

1. **Is there anything I missed?** Is there a question I should have asked? The responder may have something else he or she would like to talk about and did not because you failed to ask the question that allowed them to talk about it freely. By asking this question you are opening the floor up for any other information that they might want to add. Be careful with this question. It may open up a subject in which the discussion takes longer than the allocated remaining time. When that happens, be prepared to schedule another interview.
2. **Do you have any questions of me?** Occasionally the responder wants to know about what is going on with the project, how the information will be used, or when the requirements are going to be ready.

3. **Is there anyone else who might have something to add with whom I should talk?** This question is more important at the higher levels of the organization because usually the response identifies lower-level people to talk to (called downward references), and you gain an automatic get-an-interview-free card, which is quite valuable in organizations where no one has time to talk to you. When Mary tells you that her employee, Susan, is the person to talk to about voucher entry, you are guaranteed an interview with Susan regardless of her reluctance to talk to you when you say, "Mary suggested that I talk to you to learn about the voucher entry process."

Interview Follow-Up   Follow up any interview with an e-mail summary of the information received and any questions you have about the information or any clarifications needed. Relate the use to which the information will be put. Invite the responder to add, delete, or correct any of the information or conclusions you have reached based on that information. Leave the door open for additional interviews in the future.

Interview Issues

Interviews are not easy to conduct. Even with what may appear to be a successfully executed interview, the information you obtained may not provide the answers you are looking for. It is difficult to listen actively, prepare the next question, and keep track of what has been said so you do not repeat questions or ask for the same information, plus you must take full and complete notes of the meeting at the same time. Here are some other issues voiced by business analysts.

Assumptions   "We didn't get good requirements because there were too many assumptions."

Let us consider a scenario. The user requests an accounts payable system and commences to define the aged accounts payable report, the cash requirements listing, and the vendor payment history reports. She provides a sample of the voucher that is created from the vendor invoice, and together you draw up a screen design for the entry of the voucher information into the system.

She describes the process of posting the voucher information and gives you the general ledger accounts against which to post. You have several meetings with her to clarify the details of the process. You produce a whiz-bang system that does everything she wanted except print the checks. She is aghast. How could you do an accounts payable

system without printing the checks? You are surprised because she never mentioned it, so you thought it was like payroll where the payroll computation system is separate from the payroll check printing system. In fact, you figured the accounts payable system probably used the payroll check printing process.

The accounts payable system check writing is, of course, different. She did not tell you about the check writing because she assumed you would know about it. Everyone knows about it. At least everyone she is working with, namely all the accounts payable personnel. Having had no experience before with accounts payable, you had no idea. She assumed, since the systems belong to the information technology department, that you should know the ins and outs of the accounts payable system intimately; you assumed, since she did not tell you, that printing checks was not part of the new system.

One of the common ways to combat assumptions, both yours and theirs, is to ask more questions. A good question to ask to counter assumed assumptions is, "How do you know?" Applying Miller's law is a good guideline. Also, listen for verbal cues: "obviously," "clearly," "what they are trying to say is . . . ," "always," "never," "generally," and so forth. Behind each of these cues is an assumption.

Ownership   "The users don't participate until after the system is delivered. Then they tell us everything that is wrong with it."

Another issue is ownership. As long as the users believe that the systems are owned by IT and not by them, they will not bother to contribute completely and fully, and they will always assume you know as much, if not more, about the system than they do.

The earlier in the development cycle that the user community takes possession of any system or application and all its components, the more likely they are to make sure that the business analyst and solution team understands the system or application. When the application is not theirs, they have no vested interest in it until they have to use it.

From the start refer to everything in the problem and solution domain as belonging to the business community. Instead of writing "my requirements," you write "your (the stakeholder's) requirements." Do not ask them to review "my functional requirements specifications," ask them to review *their* document, even after you spent three grueling weeks writing it. It is not the solution team's application programs and Java code; those items belong to the voucher entry team. When you refer to the problem and solution as belonging to the business community, the business community takes ownership, has a greater interest in the outcome, and participates more fully in the production of that outcome.

🔍 **Example**
_____

In one large project a while ago, I established a fine of $5 whenever anyone on the project team referred to any part of the system we were installing in the possessive ("mine," "my," "ours," etc.), whether talking internally to other project members or to the stake-holders. When a programmer said, "I'm working on *my* program" it cost him $5.

   The team played the game so well they ratted each other out to the tune of $370 at the end of the project, which paid for a nice party. They even caught me one time referring to some reports I was writing for management as my reports. I coughed up the five bucks. So, instead of saying "I have to make changes to my program," the developers said, "I'm working on your feature or change or system." Magic happened. The users began to take possession of the system during the early prototyping sessions and were much more assertive about requesting changes, and more definitive about describing what they wanted. They discussed the upcoming system among themselves, arrived at prototyping sessions early and stayed late playing with the system, asked more penetrating questions, and generally were more concerned with the quality of the system than other organizations had been. They realized that the system was theirs and it was their responsibility to make it right for them. We were just the builders of their house. Normally the prototyping sessions were demonstrations with the developer showing the participant how the system operated or the changes made since the last session and the participant would comment. After a while the participants asked if they could work the keyboard and demonstrate it to themselves. The standard comment: "Of course, it's your system!" Even the programmers noticed the difference in the user community and found the experience to be refreshing and gratifying.

Accessing the Real Users   "We don't get good requirements because we can't get to the people who have the right information."

   This is a common complaint among business analysts. It could be a manager's decree, or corporate policy, or geographical distance, or user resistance. In any case, those who are going to use the solution are not the ones who are providing the information on how they are going to use it. Mostly, the problem is access to one person who is absent, recalcitrant, or unavailable.

   What to do: Focus on the information, not the individual. When you are unable to access your first source of information, take the request for

information to the problem owner, or anyone in authority in the organization who wants the problem solved, and ask for the information. That person may get the information for you, provide you with another source you were not aware of, or arrange for you to get the information from your intended source. In any case, you get the information necessary to continue. It is better to get the answers directly; it is also better to get the information than to stall the project waiting on an individual.

The Anti-SME   "We don't get good requirements because [management] assigns the least valuable people to meet with us."

Usually you do not have a choice about which of the process workers you talk to. You run a large risk when you simply ask the manager to identify "a couple of your users to spend a couple hours a week for a few weeks to get these requirements hammered out." You expect that the manager wants the best solution to the problem and therefore will assign his top people. This is not always the case, especially when the manager is not the one with the problem or the authority. Business line managers have many priorities and the first one is the day-to-day productivity of their staff: meeting quota, getting payroll out on time, entering all the accounts-payable vouchers accurately, and so forth. A manager is more likely to assign someone who will not drag down productivity with his absence than let their top performer step off the line to answer your questions.

What to do: Ask the manager or supervisor for the specific information you need from the process workers and let the manager decide who on the staff has that information. This gets you a good responder, and informs the manager exactly what you are looking for, which may ease some anxieties about what goes on in the information-gathering session. The manager may also recommend a meeting format with more than one process worker. Another alternative is to ask for representatives from the classes or categories of process workers you defined to segment a large process worker population.

Resistance to the Interview   "What do you do when the people you need to talk to don't want to talk to you?"

What if the interviewee does not want to be interviewed? He does not want to spend the time with you and has to because the boss told him he had to? This is clearly not the best environment to get good information.

The obvious alternative is to ask the manager for another person to interview. This may send a negative message to both the prospective responder and the manager. You want to keep the authority for process worker assignment in the hands of the manager. Handling the reluctant

responder requires an approach with more empathy. Usually behind the reluctance there is a fear of reprisal or concern about what is going to be done with the information.

What to do: Meet the situation head on and ask why the responder does not want to discuss the situation. You may not get the interview you were expecting, but you will be getting information, even when the interview consists of nothing but complaining about the work conditions or job assignment.

Another technique is to draw scenarios. Describe the situation as you see it and let the responder correct you or add to the picture. You may find the responder more at ease when filling in blanks and telling stories than with answering direct questions.

There is also a fallback position when you really do need this one person's input. After conducting a series of interviews, request a review session with all the responders to confirm your findings. During this meeting you may find that the reticent responders open up because of the peer pressure of seeing everyone else contribute without fear.

Why Bother, It Will Never Work Anyway   Part of the reason for the resistance is because the process worker has participated in these interviews to build new systems before and found the time to be wasted: Whatever requirements they asked for never came to pass, the information they gave was never acknowledged or used in any meaningful way, or their contributions were completely ignored.

What to do: When you are gathering only information and not requirements that cynicism dissipates. To ensure the cynicism and resistance continue to dissipate, break the solution into incrementally deliverable, functionality operational components that the business community can see, use, and comment upon. To complete the turnaround from resistance to support, make sure that you accept and respond to all feedback received.

Do Not Ask, Do Not Tell   "We don't get good requirements because they [the businesspeople] do not tell us everything."

There are several reasons this is the case. The process workers:

- Assume the business analyst already knows.
- Really do not know the answer (i.e., have not used the system, function, or process).
- Do not think of it at the time of the interview (i.e., an annual process or a function only performed on a rare occasion).
- Cannot come up with the words to express the answer so do not say anything (especially about quality issues).

- Are afraid what they say will be taken to upper-level management so they will not tell you what is wrong or what they have problems with.

When confronted by a business analyst demanding to know why the critical information was not included, the process worker simply responds: "You didn't ask." Keep in mind an old adage: "Users will answer any question you ask completely, but only the questions you ask." In other words, the users typically do not volunteer any information.

What to do: Asking at the end of the interview whether there is something you should have asked might prompt a responder who has information and was waiting for the question to be asked to divulge it. You can also seek a second source for the information, hold confirmation sessions, and keep the process workers in the loop so that they will remember to mention everything. Listen naively. Most of all remember that the process worker will assume you already know the information and therefore fail to tell you. Again, work with scenarios and pictures, especially if you can draw a diagram. For example, a use-case diagram, in front of them, might prompt the responders to remember more information as they try to help complete the picture.

Responders Offering Solutions   "How do we deal with customers who give us the solution and not the problem?"

"What is the best way to objectively define requirements when [the boss] has given us the solution? What do we do if the real solution isn't his?"

Nowadays, due to the sophistication of the computer users and their past experience with software development and/or process improvement, users have solid ideas what they want their systems to do. They tend to describe solutions to the problem without disclosing what the problem is. A process worker suggesting a solution without specifying the problem is tough and takes some elicitation expertise. It is worse when the manager suggests a solution. Then we get into politics.

What to do: Accept the solution as additional information. You may want to acknowledge that you are aware that the responder has given you a solution and you will file it with the rest of the solutions you have received. If the solution is somewhat unorthodox or sounds confusing, feel free to ask the responder what problem his solution is intended to solve.

Solutionizing   This issue is closely related to the previous one. The boss has presented you a solution. You realize that there are other solutions and want to explore alternatives and you are afraid of what the boss will say when you come back with a better solution that is not his.

What to do: Immediately suggest another solution, or a variation of the solution given. You might say, "That's a good idea. How about if we . . . " The solution you propose does not have to be one that will be implemented, or even considered, to get him off that one-solution track; it simply has to be another solution. If the manager continues to push his own solution, offer others. Typically the manager starts solutionizing along with you, especially if one of your ideas enhances his solution in a way he likes. Once the manager accepts an alternative approach or any variation, he is recognizing that his solution is not the only one that works, and you are free to explore alternatives without risk of political fallout.

Do Not Bother Me with the Facts; I Have Already Decided This issue may be expressed as "*We were prevented from doing requirements because [the boss] had already decided on what was wanted,*" or "*They stopped us from defining good requirements because they decided a process already exists that satisfied the requirements.*" In either case, what has happened is some upper-level or mid-level manager has already decided on the software package that they want to purchase to solve the problem. It may even be a situation where the vendor provided a software solution and the sponsor identified a problem for the software to solve. In any case, the decision has been made.

What to do: When you find yourself in this situation, you must still perform your assignment to define the solution with objectivity and integrity. In other words, do not force the solution to meet a specific predetermined goal, especially for political reasons.

The manager who made the decision may not be willing to spend any time or allow their staff to spend any time with you, so you can define a solution that has already been defined. This is where the humor me approach may work. "Listen, your solution is probably best, but I do have a job to do here. So, can you humor me? I won't take up too much time from you or your staff. I just want to make sure that when the new software is delivered we can get it operational as seamlessly as possible and avoid any issues."

At worst, you get a chance to document the problem domain and the adjustments that occur when the purchased solution is installed and operational. That way you are ahead of the game when the inevitable changes are requested from this very same manager.

Many of these issues also apply to information-gathering meetings. When you moderate or facilitate an information-gathering meeting there are other issues as well. In some organizations, usually due to a geographically disparate business community, individual interviews are hard to come by, so all the information gathering is done by meetings, and usually over the phone. Let's now move to a discussion of meetings, and how to get the most information when you have a group of people facing you, either across the table or across the wires.

Information-Gathering Meetings

An information-gathering meeting is not the same as a business meeting where the typical purpose is reporting status or solving a problem. The information-gathering meeting is conducted much the same as an interview except that you are interviewing more than one person at the same time.

Creating a situation in which a number of disparate people get together in a room and provide information to you is daunting. Since no one likes to attend meetings, there is a built-in negative bias to the proposition before the first person enters the room. There are ways of turning this situation into a gold mine of information.

Meetings during requirements definition can be used to good effect. They can be confirmation sessions or conflict resolution sessions. Meetings also speed up the process of gathering information by performing a group interview instead of several independent sessions.

One of the first things to do to make the information gathering meeting more effective is to change its name from the traditional requirements workshop. Unless you expect the results of the gathering to actually be a set of requirements ready for approval after editing, the appropriate name for the assembly is information-gathering session. Table 11.7 shows a comparison of the expectations and results of the two.

When, at the end of the meeting after you have thanked the participants for their time and the information received, the participants ask you when you will be bringing back the requirements for their approval, then you know you have made the transition from a requirements meeting mentality to an information-gathering session approach.

Group Session Preparation

Your preparation for a group session includes all the items listed for any information-gathering session, with special attention to the definition of the participants, and the definition of the type of session you are going to conduct, which is an information-gathering session.

Participants   Your information-gathering session should include only those participants who can contribute to successfully obtain the goal of the session. Typically, when calling for a meeting, you broadcast the announcement to everyone involved with the solution life cycle just in case there is someone out there who might possibly have something to say, or in case there will be political repercussions when someone was not invited. A better approach: Do not have anyone in the session that does not have a role to play or information to help you achieve your objectives.

TABLE 11.7 Comparison of Requirements Meeting and Information-Gathering Session

Requirements Workshop	Information-Gathering Session
The participants expect the goal of the meeting to be a set of requirements, so they come prepared to have their requirements included in the final result.	The participants expect that the meeting is about information transference, so the participants come expecting to provide information, not requirements.
There is likely controversy and discussion when two participants have requirements that conflict.	The politics is reduced to practically nothing because there is no contention or conflict.
Some participants, especially mid-managers, attend solely to guard their turf to make sure no requirements are included that might be adverse to their area.	Mid-managers tend not to show up because there is no decision being made at the meeting.
Some participants will be afraid to suggest anything that might be construed as a requirement for which they will be held responsible.	More participants with information might show up because they do have information to provide and are not afraid of being responsible for requirements.
Some participants who have valuable information for you may not show up because they really have not defined their requirements.	The onus and accompanying responsibility for developing requirements is taken off the shoulders of the participants and placed where it belongs, on the business analyst.
Participants may even bring a set of predetermined requirements to the table ready to do battle to get those requirements into the final document expecting that the meeting is nothing more than a political showdown with others who feel the same way.	When there is conflicting information, stories can be taken from both sides without argument since it is all information and it can be sorted out during analysis.
There is frustration when all the requirements are not defined at the end of the meeting because of time or derailment or excessive attention on one or two controversial requirements.	Instead of lists of requirements, the participants bring manuals, process descriptions, business rules, and other documents that provide information that may be needed in the meeting.

Here are some guidelines for selecting participants:

- Define a role for each participant to play. That way each participant represents a different area of the problem domain and comes prepared to provide information from that area.
- Do not have process workers in the same meeting as their supervisor or managers.
- Identify all participants and roles they will play in the invitation and send the invitation out to just the participants (do not cc: anyone else).
- Ask participants who cannot attend the meeting or may miss a portion of the meeting to designate someone else to attend.

The advantages to limiting your information-gathering sessions (and many other meetings you may have to call) are:

- There is less chance that someone who does not have a specified role can derail your meeting, or bring up topics you had not intended on covering.
- When the invited participant cannot make it, they will usually inform you ahead of time because they feel responsible for the role they were assigned. Many times they will ask someone else to take their place and your meeting can go on as scheduled.
- When you announce who is going to attend and what role each will play to all the participants, every participant feels a responsibility to represent the constituency or activity they have been assigned, and will more likely come prepared.
- You can limit the number of participants attending your meeting by asking anyone not invited to state their role (in other words, why they are attending your meeting). In one recent instance when a business analyst did this, the responder brought up an area of impact the business analyst was not aware of, so she invited him to the meeting.

Agendas   Agendas are mandatory for problem-solving meetings and for information transmission or status meetings where you are passing on information to the attendees. In an information-gathering meeting, an agenda may be counterproductive. The participants tend to arrange and focus their responses around the agenda rather than give open and honest responses to all questions. When the agenda is organized by topic, the participants will try to organize their responses into the correct bucket as dictated by the agenda. This might cause disconnected information flow and points lost between the topics. When a department or group organizes the agenda there are responses only from one group during that time, even though you really want an intermixing of information from all groups to get the holistic picture. In an information-gathering session, identify objectives rather than agenda items.

Place the overall objective of the session, which consists of the problem statement and vision, at the top, and then list the information you intend to derive from the session. Important: Do not assign time segments to the topics.

Note that in our example we have stated the information we want to obtain through this meeting. We might also list the participants. For example, Susan Miller from A/P voucher entry; Ahmed Macht from accounting; Mary Cohn, the manager of A/P; and Helen Ruiz from human resources. In this case the roles in the information gathering session are inherent in their titles. Based on the agenda, all participants will have a good idea of their responsibility in the meeting.

🔍 Example

Agenda for Accounts Payable Information-Gathering Session

The vendor database as currently constituted does not allow for enough variance in vendor terms of payment. Our goal is to have the vendor payment terms be 100 percent correct. To do this we need to understand how the various activities of the overall accounts payable process work.

We would like to achieve the following with this session:

- A complete understanding of the payable voucher process.
- A description of the type of rework that is currently occurring in the voucher entry process.
- What types of errors cause the rework and the source of those errors.
- A breakdown of activities by staff member.

Group Session Introduction

It is not important to engage in preliminary warm-up rapport building in a group session. It is, however, important to introduce the participants. When the participants know each other, perform the introductions for them, identifying each participant and the role they are to play in the session. When the participants do not know each other, have them introduce themselves *and* the roles you assigned to them in the invitation.

Group Meeting Body

As with any information-gathering session, the body of the group meeting is where the information is collected. The body of a group session is more difficult for the business analyst to deal with than an interview.

This is the tightrope that the business analyst walks during the meeting. Too much direction or control and the flow of information is reduced; too much facilitation and the meeting can get out of control, with some participants monopolizing the discussion, or responses wander off into unrelated topics carrying the discussion with them, and the objectives are not met.

To get the optimum balance between information flow and control, the business analyst needs to play two roles: facilitator and moderator (see Table 11.8). The moderator controls the meeting, ensuring the flow of information comes from the participants to you. This requires directing, controlling flow, handling the interruptions and bad meeting

TABLE 11.8 Typical Activities of Moderator and Facilitator Roles

Moderator	Facilitator
Open and close the meeting.	Ask more questions.
Determine who should speak next.	Paraphrase answers to ensure understanding by all.
Select which subjects are to be covered.	Rephrase questions to other participants to encourage participation.
Table subjects and discussion to later in the meeting.	Draw pictures and diagrams (such as use cases) to illustrate what is being discussed.
Maintain the right level of detail for the session.	Move around the room to change focus and stimulate different thinking.
Keep track of time to allow everyone an equal opportunity to speak and ensure that the meeting ends on time (or breaks are taken appropriately in a longer meeting).	Ask both left brain questions (those requiring responses based on facts and logic—how does this work?) and right brain questions (those requiring creative answers and perhaps flights of fancy—what if . . . ?)
Keep anyone from drawing conclusions that may be persistent (outlive the meeting).	Perhaps lighten the atmosphere up with humor to relax people into feeling more comfortable about providing information.
Keep the speakers relatively on point without stemming the flow of information.	Coax, cajole, and coerce answers to all questions from as many of the participants as possible.

behavior, opening and closing the session, and calling for breaks. The facilitator helps the participants increase the flow of information. This requires coercing, encouraging, advising, asking questions, and so forth. Ideally you would have two people in charge of the meeting. However, that is a rare occurrence and the individual business analyst running the meeting finds him or herself playing both roles.

Both roles are neutral and neither becomes involved in the discussion nor the transference of information. Once the introduction is over and the scene is set, the participants do all the talking and the facilitator asks questions. Neither role actually participates in the meeting. Any participation might bias the information received and the ultimate conclusions reached based on that information.

Try to be aware of which role you are playing at any time. Start off as the moderator and then assume the role of facilitator once the information is flowing. Return to the moderator role only when there is an issue during the meeting or when the time has come for close.

☑ Tip

One trick I find works well: Start the meeting as moderator from a particular place in the room or with a specific stance, for example, standing at a podium or sitting at the head of the conference table. As facilitator do something different, for example, move about the room away from the podium, go to the whiteboard or flip chart to record information, change the tone of your voice, and so on. Then whenever you need to adopt the role of moderator, move back to the original position, the podium for example. The participants will instinctively understand the difference in roles and act accordingly.

Group Session Close

The close for a group session includes the same three questions discussed earlier and also does not include a summary. In a group session it is even more important to end the session on time. While it is possible to extend the time of an interview when the responder allows it, there is no latitude with the group session. Be sure to allow time at the end for the graceful close of the meeting. Remember to thank the participants for their time and information. Remember also that the goal of the close is to ensure that all participants will willingly attend another of your information gathering sessions.

Follow-up Group Session

Construct a follow-up e-mail summary the same way you do for an interview. Send it out the next day to all participants. Do not provide a copy of the summary to anyone not in attendance. Not only is it poor form, it sends the message that comments and answers in a meeting are open for public scrutiny and that makes participants reticent about what they say. It also rewards nonattendance. When you do send out the follow-up, send the same summary of the information gathered and any conclusions drawn to all participants.

Other Elicitation Methods

Interviews and meetings are the business analyst's primary methods of collecting information. However, other methods are just as valuable. Each elicitation method evokes one or more information gathering sessions, which contain the same five stages described earlier. These approaches are summarized here.

Questionnaires and Surveys

According to the IIBA, "A survey is a means of eliciting information from many people, sometimes anonymously, in a relatively short period of time. A survey can collect information about customers, products, work practices and attitudes."[6]

Use surveys to collect information about the problem domain and to seek codified opinions on solution alternatives when the process worker population is geographically dispersed or when there is a very large population.

Observation

Observing is a valuable technique to better understand operations in the problem domain, and to confirm verbal descriptions of business processes. There are four forms of observation:

1. Passive: Observing the activities in the problem domain with no interaction with the process workers.
2. Active: Observing the activities in the problem domain and asking questions of or engaging in conversation with the process workers.
3. Participatory: Taking on the role of the process worker for a period of time and performing the activities.

4. Simulated: Observing a simulated process taking place where observing the actual process might be too difficult or dangerous, such as observing the process of flying a Boeing 777 aircraft.

The observation method can provide a view of the information flow of the business process from beginning to end and make it easier to build the business process model. Interviews and meetings give you the pieces of process that the individuals providing the information know about. Observation lets you connect the dots. If you are a business analyst assigned to a specific business area or department, then you are already engaged in observation. Whenever possible, try the participatory observation method and perform the job of one of the process workers involved in the process. It affords you great insights into the way the process and activities work within the process.

Prototyping

Prototyping is the process of creating a simulated or working model of the system that is demonstrated to the product stakeholders for confirmation. Prototyping tends to focus on one or more functions of the system and is almost always associated with user interfaces. Where there is little user interface with the system, prototyping is not as successful as other methods.

Prototyping can be done with storyboards, use cases, evolutionary or throw-away software, or an agile method, such as XP or Scrum. Regardless of the method, it typically gets you the most usable information, and is the easiest with which to work. In many cases, there is no analysis necessary: What the users see in the prototype is what they get.

Static prototyping includes diagramming methods such as use cases, story boards, and wire frames. Dynamic prototyping is done with software.

Brainstorming

Brainstorming is a highly successful method of generating targeted information. It is primarily a solution-side method used to create alternative solutions to a given problem. It can be used as an adjunct to any other information-gathering method. It is a way of collecting the maximum number of ideas on a subject from a group without considering the validity or practicality of the ideas. The goal of brainstorming is to generate numerous unevaluated responses (requirements, solutions, ideas, etc.).
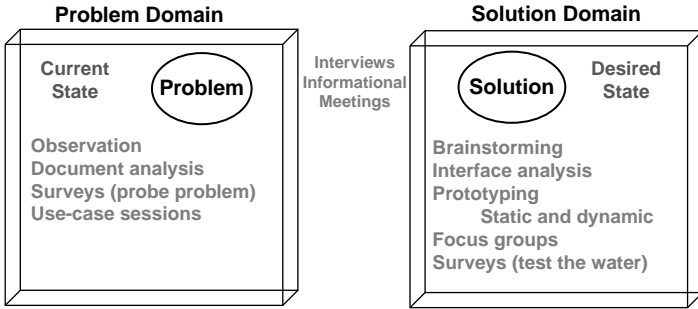
FIGURE 11.1 When to Use the Elicitation Method

When to Use Which Method

Figure 11.1 shows when each of the elicitation methods is best used. For example, prototyping is better used to demonstrate the solution and get feedback about it. Brainstorming is for producing creative solutions. On the other hand, if you cannot observe the solution, then observation is strictly a problem domain method.

Are We Done Yet?

Knowing when you are finished is tricky. Often business analysts find that ''We don't get good requirements because we don't have enough time to get enough information,'' but what is enough? Since we are not looking for an exhaustive list of requirements from the user, we can stop gathering information when we are certain we understand the problem domain well enough to start moving toward a solution.

Here are some indications that the time has come to stop eliciting information:

- When responder okays the results.
- When the model on which the information is based can be completed.
- Upon completion of a dry run or successful prototype.
- The problem owner tells you it is over.
- You have reached your objective.
- Run out of time or money.
- You have the solution(s).
- Users begin repeating themselves.
- It takes longer to get answers out of the same people.
- Different users provide the same information.

As a review of each information-gathering session, upon completion, ask:

- Did I achieve my objectives for this session?
- What did learn that is new (that I did not know before)?
- What is the value that I am getting?
- Am I still getting informational value for the time we are expending in information gathering?

This review helps you to determine:

- What questions to ask next time.
- How to improve your information-gathering techniques.
- Whether you have come to the end of information gathering, at least with this respondent.

In the end, you are never done gathering information, even after the solution document has been forwarded to the solution team and the implementation is underway. You will still be gathering information when the almost finished product is being examined by the business community and readied for delivery. There will always be a new idea, a better way, a change of mind or of heart, different management or business philosophy, an altered business or regulatory environment, or any of the millions of variations that occur in daily business life. You will be gathering information to evaluate change requests, to participate in negotiations or mediation, to resolve conflict, to present data to upper-level management for decision making.

However, gathering information is only part of the complete picture. Information is just data until it is analyzed and given meaning in context. The project manager, solution team, problem owner, business community, upper-level management, all need more than just information recorded and passed on. They need the analysis only the business analyst can supply, the analysis that discovers a problem behind a list of symptoms, and a solution in a collection of interview notes and meeting transcripts.

Let's assume we have achieved a critical mass of information. Now let us embark on the analysis of that information. First we analyze the information to produce a definition of the problem domain. We need to know where we are before we can get to where we want to be. Chapter 12 presents a process for defining the problem domain in a short amount of time even when you are totally new to the domain. Read on.

Notes

1. International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge*, version 2.0 (March 31, 2009), 53–62.

2. Ibid., 169.

3. Ibid., 53.

4. Software Engineering Institute, *Capability Maturity Model Integrated/Software.* V 1.2, 392–393.

5. Frost, "Interview," *USA Today* (January 23, 2007).

6. International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge*, 214.

Define the Problem Domain

The new standard is to try to understand the business domain before, or in parallel with, a software engineering project.
—Philip Krutchen

You cannot determine where you are going unless you know where you are. The solution document defines what is necessary to close the gap between what is and what we want it to be. To determine that gap, we need to know precisely where we are, so that we can determine the conditions that are causing the problem we need to solve. This chapter addresses the problem domain and how to define it.

Ideally, the first source of information needed to put together a solution after you have defined the real business problem should be the latest baselined solution document defining the business processes and supporting systems currently in operation. This document ostensibly describes the problem domain accurately and currently. Of course, such a document might not exist, at least in a form that reflects the current system. This is a compelling reason to keep the persistent documentation up to date. However when such a document does not exist, the first step is to create one.

There are many reasons for defining the problem domain first. You can clearly understand where the problem lives in the business processes. You can clearly see the conditions that cause the problem. You can tell what other business processes may be impacted by any solution you consider. You can define and circumscribe your focus. Certainly defining the problem

domain is extremely important. It is also something not done too often. There are a number of reasons for this lapse:

- The business analyst has been working in the problem domain for a number of years and knows the domain well enough not to have to define it first.
- The business analyst trusts the business to have defined the requirements and therefore does not feel the need to know anything else about the domain.
- There is only enough time allocated to get the basic requirements for the functionality of what is needed.
- The business analyst does not see the value in providing context to the solution.
- The stakeholders do not see the value in providing more than the minimum amount of information to describe the problem.
- The business analyst is restricted for any number of legitimate (or not) reasons from venturing past a few stakeholders.
- The project manager and solution team are waiting for the requirements so they can get started on the solution implementation, and don't want the business analyst wasting time defining the business processes.

All information gathering begins with the way things are, the current state, or the problem domain. The business analyst must have complete understanding of the current state before proceeding with defining what is necessary to solve the problem. As shown in Figure 12.1, you must define and investigate the whole problem domain before defining solutions.
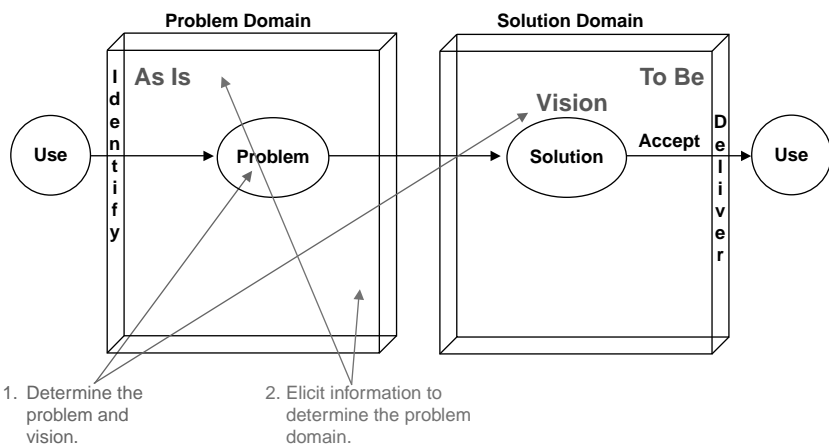
FIGURE 12.1 Defining the Problem Domain

Problem Domain Analysis

To truly understand what has to be done, you must first understand the current situation. When you define the business processes in which the problem exists you get a complete picture of the situation before determining a solution. Not only will you be able to better understand the impacts of change on other aspects of the business, you can also determine operational measurements before the change to compare with post-implementation measurements.

Except in the simplest of situations, defining the business problem alone is not enough to completely solve it. There may be other contributors to the problem that are subtle, indirect, or perhaps somewhere else in the process. There may be factors that limit the solution due to impact.

You may also discover that you cannot really get a handle on the problem based on the information you have obtained. You may find out that you need to get a clearer picture of the circumstances surrounding the problem. Sometimes it is easier to define the target business process and associated processes first so that you can see the problem and its full impact. Whether you define the problem or the problem domain first, you need to eventually define both.

As shown in Figure 12.2, the problem domain defined by the business analyst is broad rather than deep. It crosses departmental boundaries to define the business process surrounding the problem. Once the conditions that cause the problem are identified, the business analyst focuses on the solution that changes those conditions to resolve the problem. With successive iterations, and incremental deliveries, the solution is defined and implemented creating a new problem domain.
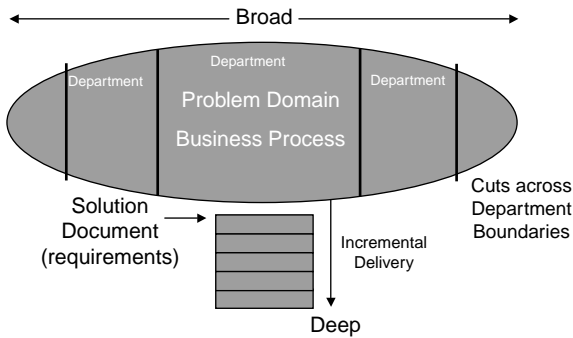
FIGURE 12.2 Problem Domain

What Is the Problem Domain?

The problem domain consists of those functional areas of the business that are affected by the problem in question or that may be impacted by a solution to that problem. The business analyst defines the problem domain as a self-contained part of the business processes, which encompasses every activity affecting or affected by the problem. The problem domain exists whether we are talking about a small change to a single Web page or an entirely new green field system that will impact the entire organization. The problem always resides somewhere within the problem domain.

Why Define the Problem Domain First?

Unless you have been working in the business unit for several years developing new systems, features, and modifications to processes, you have to spend time defining the business process in its current state. This is critically important:

- To accurately and completely determine the source of the problem.
- To fully assess the impact of any solution on other areas of the problem domain.
- To ensure complete understanding so that nothing is missed, especially in the area of nonfunctional requirements.
- To make sure there is a baseline against which regression testing will be done to ensure no collateral damage has occurred as the result of the changes.
- To give the developer a clear understanding of the whole environment beyond just the changes to be made.
- To learn what is necessary for the deployment transition.
- To identify what is not to be changed along with what will change.

A problem domain definition will minimize the issues expressed in these common statements: "We can't get good requirements because the users don't understand what's currently in use. The client doesn't understand their own processes"; and "We don't get good requirements because the system is overly complex—no one really understands it."

Point Solutions

The typical fare of the business analyst is a series of small projects that address a series of small problems that have to be fixed immediately. The problems usually end up being symptoms of a larger problem. Sometimes the efforts are similar to patching holes in a leaky dam that should be replaced as we try to keep aging legacy systems operational to avoid the cost and disruption of a full replacement.
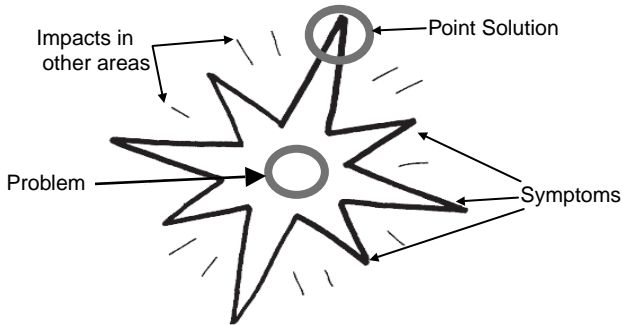
FIGURE 12.3 Point Solution

An example is shown in Figure 12.3. The problem exists somewhere in the business operations and manifests itself in many different ways. In our accounts payable story, Mary has upset vendors because their payments are not correct (two symptoms), Susan sees the overtime budget in danger because her staff is working excessive overtime (two symptoms), and Charley is not getting to happy hour on time (one symptom). There is also an impact on the general ledger accounting department that has to keep making adjustments to cover for the changes to the payments to the vendors. All of these problems emanate from the single problem that does not even seem related to any of the symptoms: the payment discount process is not flexible enough. As long as we focus on the symptoms, as one problem is fixed, new issues appear, and when those are fixed even newer ones pop up, and sometimes the old ones reappear. Until the business analyst is able to step back and look at the bigger picture, the problem may never be solved.

Gain the Holistic View

The business analyst looks at the larger picture. You apply your inductive reasoning to the environment surrounding the stated problem to discern any other problems. By looking at the whole problem domain instead of only focusing on the immediate issue, you:

- Get a wider view of the problem in context.
- See ancillary problems and issues.
- Get a better view of the impacts that may attend a given solution.
- Are able to grasp different views of the problem and the conditions that cause the problem.
- Can see the connections between requirements, design, and testing in the solution domain.

Case Study

For example, the problem domain for the accounts payable problem we defined in Chapter 8 includes all activities in the organization that use the vendor database. Once you look at that as the problem domain, instead of only focusing on vendor entry, you realize that the problem domain includes the entire accounts payable department, the purchasing department, inventory, and perhaps parts of general ledger processing. It helps eliminate this common complaint, "We don't get good requirements because no one understands the big picture."

Do No Harm

We used the analogy of the business analyst being an internist, a doctor, a medical diagnostician. The Hippocratic Oath says, in part, "keeping myself far from all intentional ill-doing." When we fail to evaluate the problem domain fully and miss a negative impact that reveals itself during production, we are just as liable as the doctor who misdiagnoses a fatal disease because a symptom was overlooked or ignored. Adopting the defensive posture of the medical practitioner, you must assume that whatever action is taken to solve a problem will result in a negative consequence somewhere.[1]

Defining the Domain

"We didn't get good requirements because we didn't fully understand the business environment."

The best way to define the problem domain is with a diagram or model. Diagramming the problem domain with workflow diagrams or other methods is also an analytical process that helps you discover what is missing or what is contradictory in the information you received.

Domain Diagram Model

The domain diagram or model is a holistic approach that circumscribes the problem domain from the outside in. It is an analytical tool to assist in determining conditions that cause the problem.

Steps to define the business process are:

1. Define the goal of the process under consideration—why the process takes place.
2. Identify the result of the process—what the process produces that satisfies the goal.
3. Identify the constituents (roles) who require that the process be executed and who receive results of the process.
4. Define the trigger that causes the process to begin and what needs to be in place for that trigger to work.
5. Identify what has to be in place before the process executes and what changes in the environment when the process is complete.
6. Identify the supporting information and mechanisms that ensure the process is successfully executed.

Figure 12.4 shows how the components described in these steps relate to one another. Each component is defined in more detail as follows:

1. **Process goal.** The process goal is what the process achieves. It is typically stated as a verb phrase. For a payroll process, the goal might be, "Pay the employees." Note that an alternate goal might be, "Pay the payroll taxes."
2. **Results.** The results define a specific output of the process that proves the process is successful or has achieved its goal (in other words, if the results are not correct or are missing, the process has failed). The results may consist of more than one artifact. Results must be persistent and are recognizable by the beneficiary as proof that a goal has been obtained, completely satisfying the requester's request. For example, in a payroll
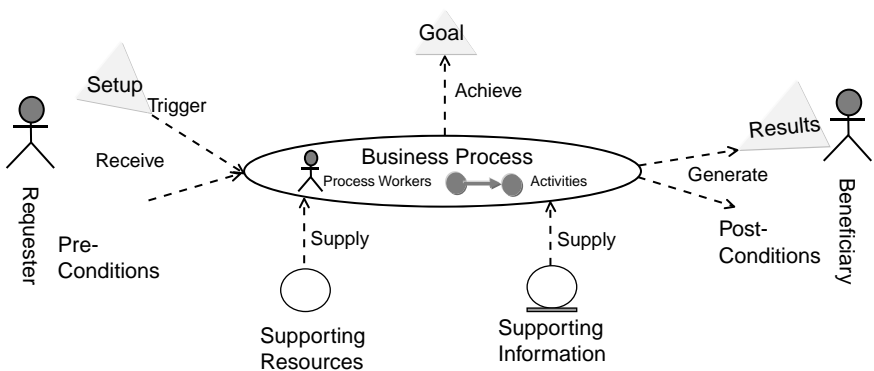
FIGURE 12.4 Domain Diagram Model

process the statements of earnings or checks are the result of the process when the goal is to pay the employees, and the transmission of payroll withholding and supporting information to the IRS is the result of the process with a goal of paying the payroll taxes.

3. **Requestor and beneficiary.** The requestor makes the request for the process to take place. The requestor is the one who has a stake in the process being executed successfully, and is stated as a role rather than a person. For example, in the payroll process, the requestor is most likely the payroll manager. Note: A requestor may also be another process.

  The beneficiary receives the results of the process; the one for whom the process is executed. The beneficiary is often the same role as the requestor. In our payroll example, the beneficiary is either the employee or the IRS, depending on the goal chosen.

4. **Trigger and setup.** The trigger is the event that causes the process to begin. It is the last thing that happens before the process begins; and launches a corresponding reactive activity inside the process. Multiple triggers may be possible (e.g., error process) but only one trigger launches the process. However, when multiple triggers are identified, you have to analyze them carefully. It may indicate multiple business processes that achieve the same goal in different ways. In our payroll example, the trigger is most likely a specific time, such as, "Every other Thursday at 2:00 A.M.," since payroll processes are usually done through automated job scheduling (rather than a payroll clerk hanging around waiting to push a button to start the process).

  The setup is an optional component which defines what must be done in preparation in order for the trigger to work, and usually refers to the data on an entry screen. For example, to use a fitting analogy: The trigger on a gun causes the shooting process to take place. However, the trigger does not invoke the process if there are no bullets. Putting the bullets in the gun is the setup. The bullets are necessary for the trigger to work. However, the bullets alone do nothing unless the trigger is pulled. There is not always a setup, as in the case of the payroll process. Our payroll clerk is not present and entering data just before 2:00 A.M. Perhaps, though, there is a job-scheduling file that has to be present that contains the pay date, last check number printed in previous payroll run, and so on, without which the payroll will not execute at 2:00 A.M. The trigger will not invoke, the process will not execute, the employees will not get paid, and there will be a payroll clerk in deep trouble.

5. **Preconditions and post-conditions (optional).** The preconditions are all data and resources necessary for the process to complete successfully that have to be in place before the first activity in the process starts. Failure to have a precondition in place means the process must be aborted at some point in the execution. For our payroll example, one

precondition is that the paycheck stock is loaded onto the payroll printer. Another might be that all new employees have been added to the payroll master files during the preceding payroll update process.

The post-conditions are all the other persistent artifacts or information created by the process as it executes. The post-conditions represent a change to the environment (a report is delivered, data is updated, e-mail notifications are sent, etc.). In the payroll example, a post-condition of the process with the goal to pay employees is the reports to the IRS, and a post-condition of the process with the goal to pay the payroll taxes would be that payroll notification is sent to the employees.

6. **Supporting resources and information (optional).** Supporting resources are the people, hardware, machinery, operations, and so on that support the process and are not intrinsically part of the process itself (in other words, not a process worker). This support does not have to be available when process starts, but must be there when needed, and may be needed in multiple activities throughout the process. A typical example is an authorizer or outside auditor checking on specific activities or outputs of a process, causing the process to stop until the authorization has been made. In our payroll process there is most likely no supporting resource since the payroll is run automatically.

Supporting information is the information that the process consumes during its lifetime without which the process must stop and wait. This information is not needed at the beginning of the process and may be input later (e.g., a file that is merged into the process after it is updated by another process). The information may be needed in multiple places throughout the process and may be transformed or changed by other processes between uses.

The difference between whether an interface is a supporting mechanism (i.e., human) or supporting information is whether the actual person is needed or just the information. For example, when a customer needs to provide a signature with the information, say for confirmation or legal purposes, then you are talking about a supporting mechanism; when you can get the customer information from anywhere and do not need to actually see the customer, you are talking about supporting information.

Inside the Process

Once the business process has been defined by the elements previously listed, then the business analyst can go inside the process to define the process workers and activities or tasks within the process.

The process workers are completely within the process, which means that their job is to perform activities that further the execution of the process

to achieve the stated goal. Process workers may work on multiple processes (even at the same time) and typically must always be available to the process and cannot be a bottleneck.

The Event Horizon

Each process activity starts with an event and ends with a transformation or output that is the event for the next activity. This is called the "event horizon" and establishes the business scope or focus of your investigation into the solution. By careful definition of the events at the boundaries of your target process, you can narrow or expand the scope of your solution.

Figure 12.5 shows an example of a problem domain diagrammed. The process defined in the example is our accounts payable (A/P) process. Due to the vagaries of the information-gathering process our information was not gathered in the order described earlier, but the information was indeed gathered.

In step 1, we defined the goal of the process ("Pay invoices to receive the maximum payment discount"). Immediately we realized the problem we have defined affects the overall goal of the accounts payable process. Because the payment terms are not entered correctly for all vendors, the maximum payment discounts are not being taken.

In step 2, we determined that the result of the A/P process (as far as the owners of the process are concerned) is to pay the vendors, which also established our beneficiary, the vendor who receives the checks. In addition the process generates the A/P journal that goes to the general ledger process, and
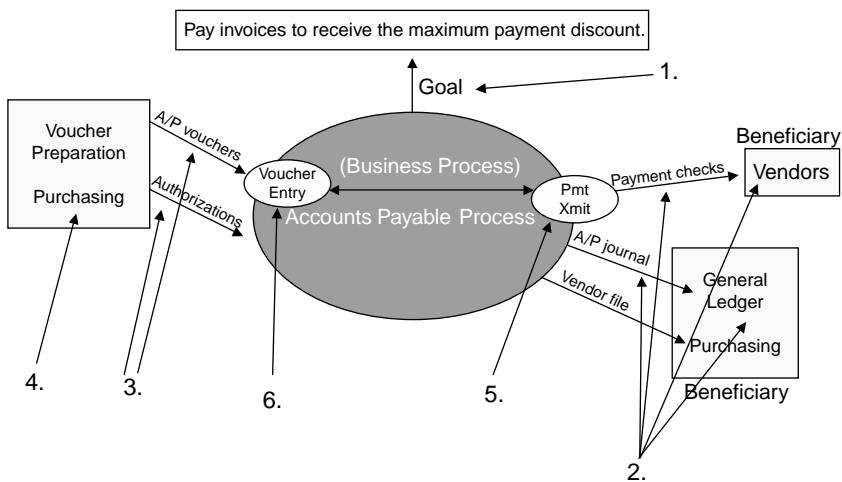
FIGURE 12.5 Domain Diagram Describing the Problem Domain (Partial Diagram)

the vendor list that goes to the purchasing department. There are other outputs as well. (The general ledger process and purchasing department, receiving some output from the A/P process become neighboring constituencies and will, at least in part, be included in our overall problem domain.)

In steps 3 and 4 we determined the trigger, which is the receipt of A/P vouchers into the A/P system. This is done in batch so the setup is the creation of the batch file. This means that the voucher entry process is a neighboring constituency feeding data to our A/P process. We also noted that for some vouchers there must be an authorization from the purchasing department for the vendor invoice to be paid. This is a precondition: no authorization, no process. We also discovered that there are no other authorizations needed once the A/P process starts and all additional information needed to complete the process is maintained within the process. This means there are no supporting mechanisms or information.

As the first steps (steps 5 and 6 in Figure 12.5) of defining the business process itself, we defined the activity within the process that produced the process result: the last activity in the process. In this case the payment transmit program ("Pmt XMit") generates the payments to the vendors. Then we defined the activity that responds to the trigger, which in this case is a voucher entry program.

The business process which is then defined encompasses all transformations in data, from the point of entering the process in one state and exiting in another (starts as a voucher and ends as an electronic payment); interactions with other data (updating the vendor, payment, and vendor history files); and algorithms and business rules that affect the process (calculation of payment discount, paying only authorized vouchers).

Changes in the Problem Domain

The problem domain may actually change size while you are gathering and analyzing the information or during analysis. You may change the domain to narrow or expand the focus by adjusting the boundaries to include or exclude activities, processes, and process workers based on the information gathered. Remember—though the problem domain may change, the stated and approved problem does not.

Here is an example:

As a result of your early investigation you discover that June, who handles voucher preparation, sorts the incoming invoices that she receives from the mailroom. While you are not sure that there is any affect or impact on her procedure, you consider that one alternative solution might include the elimination of the voucher preparation

process so the voucher entry team receives the invoices directly from the mailroom. You also wonder about how she sorts and distributes the vouchers after her department transcribes the invoices into vouchers. Perhaps you could automate that manual process and collect the voucher information in two places? You are not evaluating potential solutions at this point. To include the possibility of making changes to June's department, you expand the problem domain to include her department as shown in Figure 12.6.

Similarly your investigation shows that there is a computer process called A/P check print, a holdover from when the vendor checks were all printed on check stock and mailed. You realize that there will be no changes to the check process or the electronic transfer, so you remove that process from the problem domain as shown. Note the change to the beneficiary. The new beneficiary is the A/P check-print process rather than the vendors. The event that transmits the payment information to the A/P check-print process is now the completion of the A/P update process.

By altering the problem domain slightly you have shifted your focus to the left (toward the input side). This is valid since the problem you are solving has to do with the input of the vendor information. You do not want to dilute your focus by examining the payment layouts, the file formats of the data going to the bank for electronic transfer, the EDI (Electronic Data Interface) processes, and perhaps a manual check-writing option for vendors who do not get electronic transfers. By reducing the problem domain focus you have more time to spend solving the real problem.
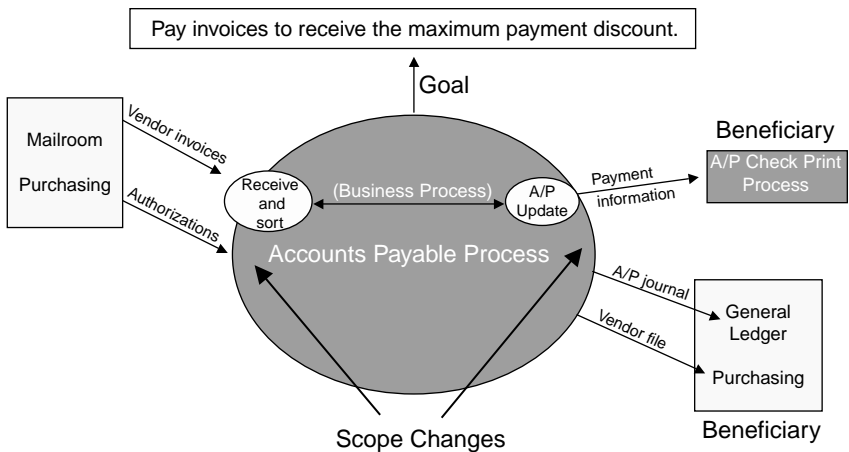
FIGURE 12.6 Expanded Problem Domain

Neighboring Constituencies

Once you have defined the target business process you need to take a look around the business process. What processes are providing the input to the target process? Where do the preconditions come from? What processes are consuming the outputs from the target business process? Where do the post-conditions go? These processes are called neighboring constituencies. And what processes share activities with the target process? These are called in-tersecting processes.

Each of the processes surrounding the target process is subject to impact when a solution is implemented. Each of the processes around the target process may have activities that can be modified or improved when the problem is solved, resulting in newly enfranchised stakeholders. If there are any indirect stakeholders around, they probably live in one of the neighboring constituencies.

You don't have to diagram each of the neighboring constituencies in detail as you do the target process. You just need to know they are there so that you can assess the impact of the various solutions you evaluate.

In Figure 12.7, the target process is the pay-the-vendor process, which is part of accounts payable. Vendor payments are initiated with the purchase goods and supplies process, which is part of purchase ordering. This process is a neighboring constituency. Other processes that feed the pay the vendor process might be the receive goods and supplies process and the return damaged goods process, which are both part of inventory, among others. On the beneficiary side of the target process is the transmit-vendor-check process, which prints and sends out the vendor payments. All of these are neighboring constituencies and may have impacts (positive or negative) from whatever changes we make to the target process.
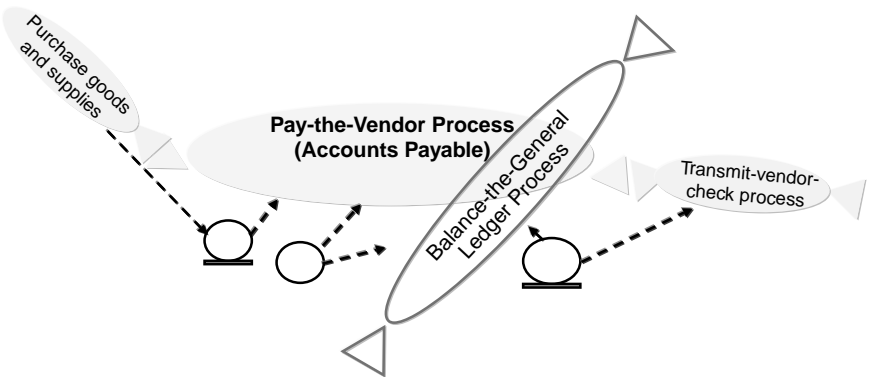
FIGURE 12.7 Neighboring Constituencies

The balance-the-general-ledger process shares some updating activities with the target process, such as posting payments, which spin off the check file and post the payment to an accounts payable journal. This process is an intersecting process. Should we make changes to the shared activities there will likely be an impact. There may also be symptoms showing up in the balance-the-general-ledger process, as a result of the problems in the target process, that may have to be addressed in our overall holistic solution.

Ancillary Benefits

You can gain more than getting a holistic view of the problem by defining the target business process at the start. As you diagram and define the business process you will discover areas that might be improved: activities that can be eliminated or revised, better ways of performing activities or tasks, methods to automate manual tasks, and so forth.

Just the act of diagramming exposes issues and problems in the business process, as process workers begin to understand, perhaps for the first time, how the overall process of which they are a part actually works.

As an example, I was working at a state agency in New York discussing a change management program with a group of budget and construction people. There was quite a bit of discussion that became heated. I was not sure what the whole discussion was about and had lost track of the overall process, so I asked for a moment to get my bearings. I stepped to the flip chart and began to draw the domain diagram described earlier in this chapter. I called the process "manage change," which was the general topic of the discussion, and asked for a goal that they decided was, "Approve a change request for construction change." Then, I asked for the result that proves the goal has been met and was told that the result is the completed change. Immediately we all saw the miscommunication that was going on. The goal was simply to approve the change, but the result of the process was the completed change. I never finished the diagram. The group more clearly defined both the goal and result by defining two processes, one to get the change request approved and the other to monitor the change through to completion. This enabled the group to complete the meeting successfully. Just the mere effort of diagramming exposed a process problem.

Change in the Problem

First of all, a problem does not change. It remains a problem until it is solved or until the conditions that cause the problem change. There are times, however, when the real problem, although analyzed thoroughly earlier, may not

truly become apparent until the problem domain is fully defined. When your definition of the problem changes as a result of information uncovered during the investigation of the problem domain, it simply means that the original problem was not defined correctly. This does not mean you made a mistake or that you failed to analyze correctly. It may simply be the result of new information that was previously hidden or unavailable. Keep yourself flexible enough to allow for the problem statement to change at this point. Do not cling to a definition of the real problem that has been superseded by new information.

When the problem changes during problem domain definition (and no later), remember to take the new problem definition to the problem owner or whoever originally approved for re-approval. Also be prepared to explain the circumstances that caused the problem definition to change.

The Essence

Once you define the problem domain, you have the basis for the solution. It is easier to define a solution by modifying the problem domain diagram than to concoct the solution from the vague and ambiguous information derived from the business community. The accuracy of the one best solution that you come up with is dependent on the accuracy of the problem domain on which it is based, so make sure you get your documentation of the problem domain confirmed by the business community.

Now you can apply the full force of your analytical abilities to the problem at hand. Having the problem domain in front of you allows you to look for elegant solutions. You may see patterns and synergies in the domain definition. You may see anomalies and inefficiencies that need to be addressed. You can see the steps and activities that are no longer necessary. You can determine activities that do not contribute to the overall goal of the process. You can craft your solution to reduce or eliminate waste, delay, inefficiency, and redundancy. Your solution will be simple, complete, and implementable. In the next chapter we talk about the process of analyzing the problem domain and the information gathered during elicitation and investigation to determine one or more solutions to the business problem.

Note

1. Peter De Jager, ''Tempering Traumatic Transitions,'' www.technobility.com (accessed October 2, 2008).

Determine the Solution

Analysis is 90 percent business driven and 10 percent technically driven.

—Lois Zells

The business analyst reaches into his or her toolbox and brings forth the right tool to analyze the information that has just been elicited. The business analyst may categorize, model, filter, diagram, create a matrix, or simply apply critical thinking. The end result is a better understanding of the problem domain and the solutions that will solve the problem, as well as a bunch of new questions to ask.

Figure 13.1 shows analysis as a third activity. This for most of you is where the fun happens. You are solving the problem. You are applying your innate analytical skills to the mass of information you have collected and are producing the "One Best Solution" to the business problem. This is the time of "ah-ha" moments, head slaps, finger snaps, sparks of insight that produce the elegant, simple solution. This is the analysis part of business analysis.

The Accordion Effect

To a business analyst, information is like an accordion—the quantity always goes up and down. Investigation and elicitation increase the quantity of information. Analysis reduces the quantity of information. And, as you reduce the amount of information gained through elicitation, holes in the information will appear; contradictions will materialize, assumptions will be exposed, and more information will be necessary to resolve issues. This means more information is elicited and the quantity of information grows again.
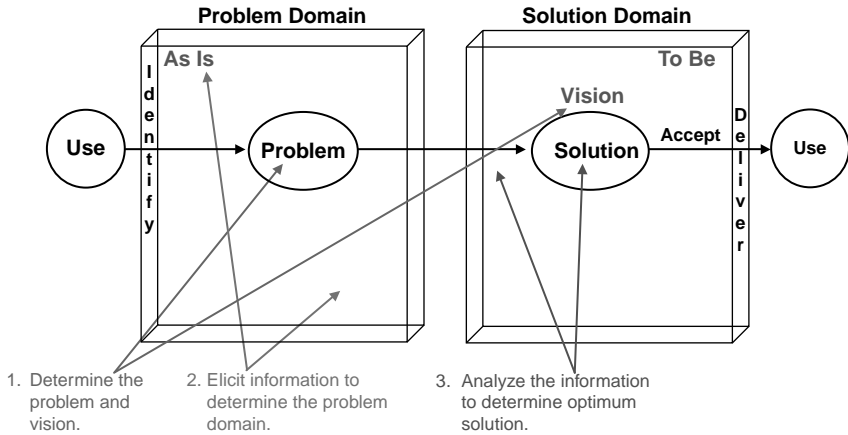
FIGURE 13.1 Analyze the Information to Determine the Solution

Each time you iterate, some information will be complete and requirements can be defined. Eventually, when there is enough information, the solution is complete. That's when the solution definition process stops. Either that or when time runs out, whichever comes first. With an agile software development approach, each iteration produces working software from the requirements defined for that iteration, which moves the solution team closer to the final, complete solution.

Tools and Techniques

''Are there any tools for business modeling and, if so, which ones should business analysts use?''

There are many analytical tools and techniques developed over the years for business analysts or which business analysts have adopted. The tools are primarily software packages that provide diagramming assistance such as Visio or Smart Draw, or software development life-cycle suites such as Rational, Together, or System Architect. The processes or standards of the organization may dictate the tools and techniques used by the business analyst. As a general rule, the business analyst uses the diagramming methods that IT defines as standards. Beyond that, the choice is driven by the exigencies of the problem to be solved, and the personal preferences of the business analysts solving the problem.

Once you have the problem domain defined, analyze the information to determine the conditions that cause the problem. It may be just two use

cases out of twenty, a small area of functionality in the overall domain, a process step or two, or it may be the entire domain. Your focus for the rest of the solution development is going to be on changing or eliminating those conditions.

Categorize the Information

The purpose of categorization is not just to allocate information or requirements into buckets. The goal is to identify information or requirements that *cannot* be classified or categorized, such as requirements or information that is so unclear, vague, ambiguous, or abstract that it could go into multiple categories or none at all. Another purpose for categorization is to help identify where there is missing information. For example, sorting all nonfunctional requirements into nonfunctional classes (see Appendix F) may expose the fact that you have nothing at all defined for one or more categories.

Select the categories that make sense in context of the problem to be solved. You may find that you need to do several categorizations to complete the analysis. Some example categories are:

- Scope—in or out of scope.
- Functional or nonfunctional.
- Specific nonfunctional categories.
- Business, system, or user.
- Functional goals.
- Business objectives.
- Organizational entity within the problem domain.
- Function or constraint, and so forth.

    Some of the purposes of categorization up-front include:

- Breaking the overall holistic view of the solution into functional components to narrow the focus.
- Re-scoping the problem domain.
- Identifying conditions of the problem within the problem domain.
- Creating functional goals for incremental delivery.
- Segmenting the overall problem for purposes of elicitation.

Prioritize the Information

The resulting information and/or requirements should be prioritized. It may be helpful to get a feeling for both the business and technical priorities for each group or category of functional and nonfunctional requirements. In

both cases, it is a good idea to get a reading on how important the feature or requirement is to the overall system and operation, and to determine the impact if the feature or requirement was omitted from the final product delivery in the first deliverable.

As a guiding rule, all requirements that go to solving the problem should be high priority.

Perform Event Analysis

Most process workers perform their assigned activities in a somewhat discretionary fashion. They perform a specific activity when a certain event occurs. Charley in accounts payable only enters the voucher information when a voucher is present. The event that causes him to perform his assigned activity is the appearance on his desk of the stack of payable vouchers to be entered. Should Charley come in early, before June places the vouchers on his desk, Charley has nothing to do. When June puts the stack on his desk, that event causes Charley to swing into action and start hitting the keyboard. The event that causes Charley to stop his activity may be either the absence of any more vouchers on his desk, or the start of happy hour, whichever comes first. At happy hour there is a different series of events.

A business process flows from initiation to completion by a series of activities that are responses to events. In addition to identifying the events, analyze responses to the events:

- What is the complete response to an event stimulus?
- Does the response react appropriately to the event?
- Is it the response that you would expect?
- Are there any activities that occur for which an event cannot be identified?
- Are there any events that occur that do not cause an associated response?
- Is the response a direct or indirect action resulting from the event?

Event analysis provides two views: a more definitive view of the problem domain and a clearer picture of the impact of changing activities in a business process.

Analyze Stories

During investigation you gather a lot of stories. Stories are a great source of real information containing descriptions and anecdotes about the way different process workers perform their activities, use the system, and the problems they might be having.

Stories[1] have specific ingredients. Analyzing each story to define these ingredients can help identify the conditions that cause the problem and illustrate patterns and commonalities that can be turned into solutions. The analysis will also identify aspects of the current operation that must remain the same when the problem is solved.

Each story has the following components:

- Agent—the people in the story (actors in use cases).
- Predicament—the problem that the actors are trying to solve (the goal in use cases).
- Intentions—what the actors plan to do.
- Actions—what is done.
- Objects—what the actors use: tools, techniques, and so on.
- Causality—effects (intended and unintended) of carrying out the actions.
- Context—details surrounding the actors and the actions.
- Surprise—unexpected events in the story.

When the product stakeholders describe the problem in stories, you get better results by describing the solution as stories as well. To be reviewed satisfactorily, the stories should contain the components just listed. The story should also have these characteristics:

- Plausibility—believability, the audience accepts each step.
- Consistency.
- Completeness.
- Economy—the story is complete without getting too inclusive.
- Uniqueness—the story is not open to alternative explanations.

When analyzing a story from the process workers, consider each of these factors. For example, is the story complete? Does it cover all activities necessary to solve the intended problem? If not, where is the information to complete the story?

When there are different ways of performing an activity, clarify that the way identified in the story by the process worker is, in fact, the only way that worker does that activity. The alternate ways might be done under different conditions, making each a different story.

When you are collecting information in the form of use cases, or user stories, or analyzing with use cases, you are dealing with stories. Each use case represents a unique story, as does each alternate flow. Each use case or user story can be analyzed using the criteria previously listed.

When you prepare a solution document, you are preparing a story. The story should also comply with the criteria listed previously. The solution

document should basically state, "Under these conditions, the following will happen . . . , and that solves the business problem."

✓ **Tip**

Include Stakeholders in the Analysis

While much of the analysis of information collected will be done alone or with other business analysts, stakeholders should not be excluded from the analysis process. It is unwise to wait until you have a solution embodied in a complete set of requirements for their approval or acceptance. Expose the stakeholders to the analytical process as you perform it so they can provide insights to the solution and even corrections to your definition of the problem domain. How? By making the whole process iterative. Gather information, confirm the information with the sources, analyze the information to produce conclusions, confirm the conclusions with the sources, gather more information and continue around again until you have the complete solution. Scott Ambler goes further to suggest that the process workers be involved in the diagramming activities gathered around the white board with the analysts or developers.[2]

Business Rules

"We didn't get good requirements because the business rules were not specified."

Business rules consist of "Terms expressing business concepts; facts making assertions about these concepts; rules constraining and supporting these facts."[3] Business rules exist in all facets of business life: policies and procedures manuals, security policy, HR policies, laws and regulations, tradition and culture, and what the boss said. "A business rule is a specific, actionable, testable directive that is under the control of an organization and that supports a business policy."[4]

Business rules are typically separate standalone statements. Examine business rules to determine whether they are a part of the solution. Remember that in many cases the business rules are constraints on the solution. A solution has to be developed that does not violate or compromise specific business rules.

Identify the source of the business rule: a regulation, an accounting principle, the policies and procedures manual, standard operating

procedure, tradition, an ingrained computer system, the boss. This defines whether the business rule can be changed or modified.

Determine the reason for the rule. There is, or was, a reason for all rules. The reason may no longer be valid. When the business rule is valid and applicable, document it separately from any other diagram or model used to describe the problem domain. Each business rule should be uniquely identified.

This is a problem domain issue only. We are examining the business rules that exist as part of the current situation. The changed business process and/or system may invalidate existing business rules, augment or redefine existing business rules, and/or add new business rules. Not only should these affected rules be verified with the business community and management, but it will require extra attention during the transition period (Chapter 17).

Assumptions

Assumptions occur during analysis just as they did during elicitation. Assumptions state the unconfirmed information on which various conclusions in the product scope, definition of the problem domain, or definition of the solution are based. Assumptions are conditions, events, or circumstances about which we are unsure and we need to make a decision based on the information we have at the time to move forward. With more information we can be more sure. Essentially, you are saying, "Based on these assumptions we are concluding the following . . . " When the assumptions are different, the conclusions will be different. Underlying the inclusion of assumptions is the implicit demand that the reader—in most cases the user community—will raise an objection or provide information that will disabuse the author of a reliance on the assumption should the assumption be incorrect in any way. Receiving no such correction, the assumption becomes a fact.

Many times assumptions are purposeful, made by users or stakeholders to obfuscate. Sometimes a user is not able to be specific and makes assumptive statements to cover lack of knowledge or inability to be exact or precise. Other times, the user does not want to take responsibility for making a decision about the requirement and purposefully makes information vague and open to interpretation, so that when questioned or challenged he can always claim he was misunderstood. And still other times the users do not want to take the time and effort to ferret out the specifics that are necessary to produce the needed results, so they assume you will do it for them.

> ☑ **Tip**
>
> There are some common phrases that indicate an assumption is being made beyond, "I assume that . . . ." Phrases like "Obviously . . . " "It's clear that . . . " and "What they really mean is . . . " are all good indications that an assumption follows. Even though we've been told that assuming is bad, assumptions are perfectly all right as long as they are documented. The primary culprit of poor system quality is the insidious assumption that goes undetected until production. The best antidote to assumptions is asking more questions and analyzing thoroughly.

Analysis is a continual iterative process alternating between investigating and then analyzing that information. The more information you have, the better your analysis can be. And, the more you analyze, the more information you need.

I prefer drawing to talking. Drawing is faster, and leaves less room for lies.

—Le Corbusier, architect

Modeling

"How does the time spent in business process modeling help me? Do I need to know how to do all the different types of models, like entity-relationship diagrams?"

The business analyst models to understand and communicate. Modeling allows us to better understand the problem domain and to communicate an understanding of it effectively to others, including both the business community and the solution team. In a model, existing concepts and new requirements can be seen logically related to each other.

Define and describe your solution using a model of that solution. The model can be diagrams, a prototype, or a set of requirements. A solution document is a model of the solution. The model describes the flow of work both inside and outside the computer system. While a systems analyst diagrams the interactions with the computer systems and software, the business analyst puts that interaction into the context of the overall business workflow.

Models are especially effective to:

- Show the relationships between entities in a structure.
- Define the relationships among segments of a solution.

- Show the order of event occurrences, especially if some of those events occur in parallel.
- Facilitate communications during elicitation and encourage more information flow.
- Simplify and filter out the noise.
- Define a common language (graphical and textual).
- Communicate among technology and business stakeholders facilitating collaboration.
- Promote progressive elaboration.
- Present multiple perspectives on the system.
- Transition from problem definition to solution design.
- Systematically define business goals or objectives down through appropriate levels of solution development.
- Represent required functionality (what is needed) in a way that facilitates the detailed design (how that capability will be achieved).

Using Diagrams to Analyze the Information   The information provided by the business serves as the basis for the diagrams. Everything learned about the business process should be included in the diagram, and nothing more. Diagramming the problem and/or solution domain consists of a three-step process:

1. Analyze the information you have gathered, eliminating irrelevancies, redundancies, and nonessential information.
2. Prepare a diagram from the remaining information that depicts some aspect of the domain.
3. Analyze the diagram you created (when complete) for adherence to the diagram's rules.

There are a number of possible diagrams a business analyst can use to model the problem and solution domains. In general these techniques are divided into three categories: the data model that defines the structure of the process or system; the process model that describes the flow of activities through the process or system; and the behavior model that shows the interactions between the process workers and the system or process.

"Which modeling technique should I use and when should I use it?"

Role of Data Modeling   The entity-relationship diagram (ERD) is a domain-modeling device to ensure the business analyst understands the data structures that the customer uses. Most business analyst projects are on existing systems or on new systems that use existing data structures (such as creating a new insurance product that uses the same customer, rating, and actuarial

databases). The business analyst uses the data model to help understand the data the customer needs and to define the changes, if any, to that data.

The business analyst does not typically create a data model, although there are organizations where it is a common practice. Data modeling is generally the purview of the database administrator or systems designer. One business analyst told me that when she joined her first company she took a five-week course in which the main emphasis was on data modeling. She said that in the several years she remained with the company as a business analyst she never saw another data model.

Even if a corporate data model exists, it is a good idea for the business analyst to sketch out a rough model of the data needed for an enhancement or new feature or new system. The process of creating the diagram is an analytical process that will expose missing information, assumptions, ambiguities, and more. At a consulting company in Virginia, every business analyst had the current system entity-relationship diagram on their wall, or stuck to their whiteboard. While a data model or entity-relationship diagram may not be an effective use of time for the typical business analyst, it is valuable to create a data element list of all data needed for a new function that includes the element name, description, attributes, and perhaps a location (table).

What is important and mandatory in any business solution is a list of any new entities needed for the solution and any new or changed attributes. This list is typically passed to the database administrators or specialists for inclusion in the overall data model.

Role of Process Modeling   Processes knit together all the activities and tasks of each process worker. Everything that is done is part of a process. As Edward Deming said, "If you can't describe what you are doing as a process, you don't know what you are doing."

Common process modeling techniques include data flow diagrams (DFD), flow charts, work flow diagrams, and UML activity diagrams. All are easy to understand and can be developed with process worker involvement.

Role of Behavior Modeling   When I started out and for many years thereafter, there was no need for behavior modeling because there was no behavior. The users did what we told them when they used our computers. They answered the question at the bottom of a scrolled list of directives that we programmed, and entered the data when and where we told them. They had no other options. Nowadays there are multitasking, windows, Web pages, and user sophistication. What the user is doing, when the user is doing it, and why the user is doing it become much more critical in the overall success of the solution. Users want friendly and intuitive interfaces that

allow them flexibility in their interaction with the computer system. Behavior modeling is used to describe that interaction.

The primary tool for modeling behavior is the use case. The use case description or narrative describes a conversation between the actor (business role that uses the system in this process) and the computer system the actor is accessing. The actor performs actions and the system responds accordingly. When the system responds in this way, the actor then does this. And so forth. The use case also defines the preconditions for each process and the post-conditions (what has changed in the environment when the process is completed successfully). The use case describes alternative ways of achieving the same results, and lists all the exceptions that might occur during the execution of the process and what happens as a result of each exception.

Gap Analysis

The gap exists between the business analyst's definition of the problem domain and the vision of what the problem domain should look like. Gap analysis identifies the conditions that cause the problem, and in a general sense what must be done to change those conditions, eliminate the processes causing the problem, or add functions, tasks, or activities to the process to overcome the problem.

As shown in Figure 13.2, the gap is what exists between the problem and vision or solution, between the "as is" and the "to be." When you define the gap, we can understand what is needed to fill it and document that need as a solution document or requirements. You will find it considerably easier to identify the gap and what makes up the gap when using models of the problem domain and the intended solution rather than depending on textual representation of both.
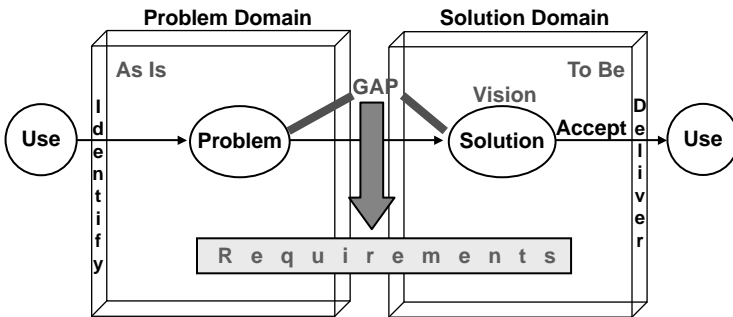
FIGURE 13.2 The Gap between Problem and Solution Domains

Defining capabilities first is a convenient way of moving from gap definition to functional requirements. Define what capabilities the current process needs to possess to remove the conditions causing the problem. This is particularly effective with common cause problems. Once all the capabilities have been defined (what you might call high-level requirements, features, functions, or user stories), each capability is elaborated into specific detailed requirements.

Determining the One Best Solution

"As business analysts aren't we supposed to be producing the best solution to the problem? Our management, and sometimes the project team, changes our solutions and not always for the better."

Ultimately, no matter how many potential valid solutions there are to the problem, and there must be at least two, there is only one best solution. The one best solution is the solution that is implemented. The solution document has the following attributes:

- Describes what is to be done to solve the problem.
- Defines the solution accurately.
- Describes the solution completely and in a manner understandable by all the stakeholders.
- Can be understood and accepted by the solution team.
- Guides the design and detail specifications and development of the solution in test and implementation.
- Identifies criteria for evaluating design decisions.
- Is independent of platform or technology.
- Meets the primary quality criteria of the business.

The additional criteria below may be applied to determining the one best solution by the business or management:

- The cheapest.
- The fastest to implement.
- The most cost effective.
- The easiest for the users to learn and to adapt to.
- The easiest to implement.
- The most politically correct.

The theory: The solution should also be independent of cost and time constraints (except for product deadlines). By this I do not mean

that the business analyst suggests a million-dollar solution knowing there is a hundred-dollar budget. I mean that the business analyst cannot let a project budget or project deadline force an invalid solution, one that will not completely solve the problem. The solution should, of course, be the most efficient way to solve the problem without any gilding or extraneous features. If the best the business analyst can do is a solution that will take the project team 14 months to complete and the project deadline is in nine months, it is time for negotiation, not cutting essential requirements that cause the project to fail.

The practice: In real life practice, many times the business analyst is forced to come up with a solution that must fit within prespecified project constraints, including technological limitations. When that happens you might find it easier to start with the one best solution that is independent of budget and time, and then, in conjunction with the process workers and problem owner, trim the solution down until it fits. It may happen that business management may decide that your solution is so good that they will come up with the additional budget, time, or both to implement your solution.

Constraining the Solution

"The system analysts jump down our throats when we try to give them requirements that have too much detail in them. We're just recording what the users want and what they tell us. But the system analysts tell us we're overstepping our boundaries as business analysts."

Many business analysts believe that they must be very specific and detailed in their definition of the solution. They want to leave nothing to chance, and certainly nothing to the imagination of the designers or developers. This is particularly true of business analysts who were system analysts in a prior life. While these business analysts may be keeping faith with the business community, they start running afoul of the solution team.

Reduce the universe of possibilities in the solution scope to those that will benefit the business and direct the solution team to the right technical solution. Do this by specifying constraints (like security, availability, business rules, etc.) that limit the number of options. Do not constrain the solution space any more than that. The systems analysts and designers must be free to evaluate and choose among trade-offs and produce the appropriate solution. The solution you specify cannot restrict the designers so only one solution is available to them.

Stop Analyzing, Already

"How do we know when we are done?"

The analysis process is the business analyst's forte, and it is also the business analyst's vulnerability. The result of the analysis, the solution document, is visible to all as is the resulting implementation of the solution. Make a mistake—overlook a piece of vital information, draw a conclusion based on an assumption, over-constrain the solution, include ambiguity—and the source of the mistake is well-known: you.

For many of us, analysis is why we are in the business. We go through elicitation for the express purpose of acquiring the information that we can then analyze. In fact, we find ourselves in such a rush to analyze and define the solution that we often start analyzing while we are eliciting during an information gathering session. We like to see the solution developing out of random data and desires. We love the feeling on that rare occasion when it all fits together and we discover a solution that satisfies all parties and all constituencies. We like it so much sometimes it is hard to stop. When we keep focused on the problem and getting the problem solved, we know that we do not stop until we have solved the problem, and when we solve the problem, we stop.

Confirmation

"The problem we have with getting good requirements is that nobody wants to validate anything. We can't get agreement about any of it. We can't get a decision out of [upper-level managers]."

As a rule, no requirement should be defined from information that has not been confirmed or verified. The exception, of course, is when the problem is for an individual stakeholder who is the only user of the solution.

Continuous Confirmation

Figure 13.3 shows the concept of continuous confirmation. Most business analysts operate on this basis as a normal part of their job. During elicitation, the business analyst confirms the information obtained with the source to be sure there was correct understanding. This is a constant effort. After each information gathering session you send out a thank-you note to the participants that also functions as a confirmation. As you define the problem domain you continually check with the process workers to ensure that your understanding of the activities and flow of the work in the target business
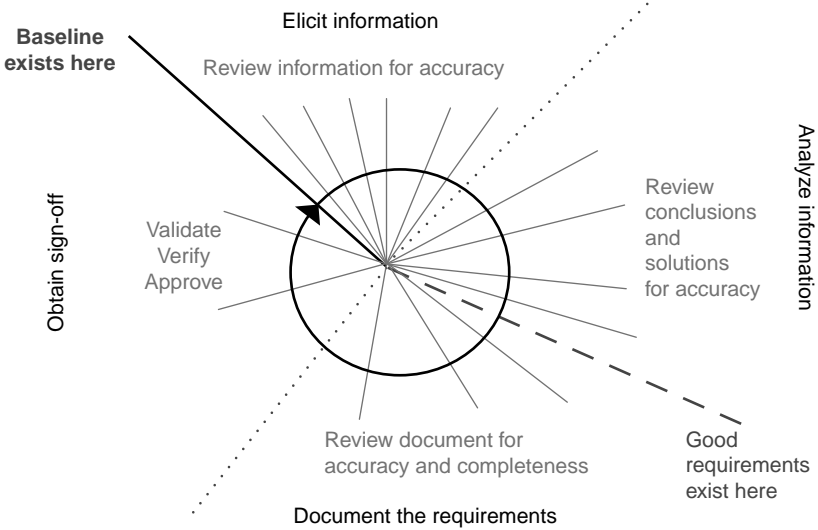
FIGURE 13.3 Continuous Confirmation

process is correct. A question while passing a stakeholder in the corridor to check on a single piece of information is a form of confirmation. The point is that confirming the information you have received and your understanding of it should not be a formal event at the end of the solution definition process.

During analysis the business analyst confirms his or her conclusions based on that information, asking in general, "If we do this, will it solve your problem?" As you analyze the information and come to solutions to various parts of the business problem, check those solutions with the process workers and stakeholders. Don't wait until the entire solution has been defined. Get confirmation for each screen, each new activity, each new algorithm or calculation, each new idea. Some solutions may be thrown out and others slightly adjusted, and some may be accepted as you have conceived them. Regardless of the response, positive or negative, it is easier to review and confirm small incremental pieces of the solution in an ongoing fashion than to hold off until the end of the project, easier for all concerned.

Confirmation is not approval. Only authorized personnel can approve. Confirmation is getting an okay from the stakeholders that the solution you are developing is good—it will do the job for them. And you do this as you develop it, not at the end. When you have a user-interface screen defined, or a new process developed, run it by the users for their confirmation. It's easier to get low-level confirmation on a continuing basis, than try to get the entire solution document confirmed and approved at the same time. As a

bonus, your constant confirmation efforts will get the stakeholders and business more involved in the solution you are developing, which means the solution will be embraced with more enthusiasm than one thrust on them as the deadline approaches.

The confirmation process does add a bit of time to the overall investigation and elicitation process. It is better in the long run to spend the time upfront asking a few more questions and discovering misunderstandings than to incorporate the misunderstanding in a delivered product and discover the mistake after it is in production or in the hands of the customer.

A requirements document that's reviewed and accepted by your stakeholders is effectively a contract that you've negotiated—a contract that defines what the stakeholders expect to receive.

—Scott Ambler

Reviews and Walk-Throughs

We all have practiced the classic method of conducting a review or walk-through: Send the document to the reviewers in advance and then meet to review. The business analyst (or author of the document) reads the document, asking the reviewers to interrupt him when they have a comment or question, or holds comments until the end of a section or chapter. Of course, the reviewers who know the business analyst is going to read the document to them have no incentive to read it in advance, making the review itself more superficial. Also, there is a tendency to mentally drift off while hearing the scintillating document being read by an even more scintillating reader, especially when, as children, we were read to sleep. This approach:

- Loses the review part of the review, since the participant will be giving the business analyst their initial reaction to the material rather than the result of thinking about it for a while.
- Reduces the possibility that participants may talk about the material before the review session, thus increasing the volume and value of the feedback.
- Raises the possibility of lost attention during the meeting when the business analyst does not read or paraphrase well.
- Confirms the business analyst's understanding of the document rather than what is actually written.

Paraphrasing the document is an alternative to reading the document in a review meeting. However, the business analyst may provide interpretation of material that does not match the content of the document, and the participants will be approving the verbal interpretation or explanation of the

content rather than the actual document. The solution team bases their development activities on the document, not the explanation that was reviewed.

Participant Paraphrasing

A better approach is to let the participants paraphrase the solution document back to you. This gives you a chance to understand how others interpret what you have written.

There are four potential outcomes when the participant paraphrases the solution document (instead of the business analyst), all of them positive:

1. The participant paraphrases it exactly as the business analyst intended it in the document and the participants approve it that way.
2. The participant paraphrases it exactly as the business analyst intended it in the document and the participants disapprove the representation and correct the document.
3. The participant paraphrases it differently from what the business analyst intended and the participants approve the participant's paraphrasing. When this happens, the document must be changed to reflect the paraphrasing.
4. The participant paraphrases it differently from what the business analyst intended and the participants disapprove of the paraphrasing. The business analyst then ascertains what needs to be corrected and changes the document accordingly.

Only when the participants paraphrase the document exactly the way you intended it and the participants confirm its correctness does the document remain unchanged.

Checkpoint Beta

Prior to committing the solution to the final, formal solution document and obtaining final confirmation of the business solution, run the document and associated materials by the development team. Schedule a meeting to be attended by the project manager, systems analyst, database administrator, and anyone else who is involved in solution implementation. Select a time to have the meeting where you know that the likelihood of significant change to the business solution is small and the solution is relatively stable. This meeting has three purposes:

1. Provide a heads-up for the solution team before the final approval is given.
2. Identify any feasibility issues that may be embedded in the solution. Technical feasibility is the primary concern.
3. Confirm with the project team and project manager that the solution will fit within the project constraints; often the project deadline and budget have already been defined.

This does not constitute a formal review; signatures are not necessary. Also, it is not necessary to formally hand out the document to the solution team since it has not officially been approved. Do not bother with a PowerPoint presentation. This is simply an informal checkpoint in the process to make sure you have not missed something or made an incorrect assumption about the technical environment.

This meeting is also a good way of establishing or keeping good relationships with the solution team. At this point you are representing the business community and simply saying, "This is what we want to do to solve our business problem. What do you think?"

When there are no more questions in the information-gathering plan, and you have evaluated all the solutions and selected the best one through analysis, you are ready to record the solution in some form of persistent document. The documenting effort is analytical and may require even more questions answered. The culmination of the analysis is the solution document that provides the blueprint to the solution team for the implementation of the solution to the business problem.

In agile approaches a Checkpoint Beta is not necessary. The solution team is involved with the definition of the solution and the functions and requirements that make up the solution. The information passed in a Checkpoint Beta is passed continuously throughout the solution-definition process.

Whether the approach you are using is agile, linear, or some combination, at some point the solution must be documented. In the next chapter we review various options to document the solution.

Notes

1. Gary Klein, *Sources of Power: How People Make Decisions* (Cambridge, MA: MIT Press, 1998), 177–178.

2. See www.ambysoft.com.

3. www.businessrulesgroup.org/brmanifesto.htm.

4. International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge*, version 2.0 (March 31, 2009), 159.

Write the Solution Document

Poor requirements definition is the root cause of bad software.
—Carey Schwaber, Forrester Research

The real work of defining the solution to the business problem is in the elicitation to gather the information on which to base the analysis and the analysis that produces the solution. Writing it down is almost anti-climatic. However, when the documented version of the solution is poorly written, ambiguous, redundant, imprecise, untestable, verbose, and generally unreadable, then all the work done to create the best solution ever is for naught. The solution when implemented will still be wrong.

The heavy lifting has been done at this point. We have a solution that the product stakeholders have determined is good: When we do everything that is stated in the solution, the problem is solved. The solution may exist in an informal format—yellow-lined paper, notes, sketches—and now it is time to convert all of it to the formal, persistent, formatted, official solution document.

The Value of Documentation

Voluminous documentation is part of the problem, not part of the solution.
—Tom DeMarco, PeopleWare

''We spend a lot of time documenting the system for the developers and then they just do whatever they want. They even go over and talk to the users and create stuff that is not in our documentation.''

Many business analysts feel that their job is about writing documentation and nothing else. It is easy to draw that conclusion. In truth, there is no role of documenter. Documenting findings and results is a part of a business analyst's other roles. It is the proof that the role has been played. We document the analysis to show that we have done the analysis. We document the requirements to show we have arrived at a stable solution to the business problem. Documentation is a means for recording completed communication. That is all.

There are two types of documentation—persistent and transitory:

1. Persistent is permanent and remains after the solution is completed.
2. Transitory is for the purpose of providing a temporary written record of some part of the process. You throw away transitory documentation when it has served its purpose.

Your real work is the elicitation and, especially, the analysis that creates the documentation, not the documents produced. Whatever we write down to assist us with elicitation and analysis—interview notes, diagrams on the whiteboard, user interface screen mock-ups on yellow-lined paper, user stories on index cards, and so forth—is transitory and can be discarded once it has served its purpose.

Persistent documentation has considerable importance in some quarters, as it should. Regulations, external system interfaces, corporate policy, and other extra-project requirements demand written documentation. Those persistent written records may constitute a large number of documents that a business analyst may have to produce or at least to endure.

"We seem to be spending all our time writing requirements documentation, like business requirements documents and functional requirements specifications. Then the users or the developers tell us it's wrong."

While a considerable amount of communication in business is done through formal documents, most of your communication should be oral and informal, that way there is no major consequence for getting something wrong—there are no documents to fix or rewrite. The business analyst has to determine how much will be done on a given project with formal artifacts, and how much will be done with informal communication. What considerations would there be in making that decision, besides corporate policy? There are positive and negative reasons for persistent documentation as described in Table 14.1.

TABLE 14.1 Persistent Documentation

Why should we document?	What are the concerns with documenting?
Regulations and legal requirements such as Sarbanes-Oxley.	Documentation is an attempt to forestall problems.
Maintain history of product and project.	It is only a reaction to previous problems.
Produce a permanent record.	To cover one's posterior (not necessarily the business analyst's).
Most people think clearer when expressing their thoughts in the permanence of documentation.	Goal becomes creating a document rather than communicating.
Provide evidence of testing, changes, auditing, completion of phases or tasks.	Additional documentation may be an attempt to slow down change—this could be purposeful to allow more time for consideration and evaluation or it could simply be heel dragging.
Facilitates asynchronous analysis over long distances and periods of time	Costly to create and maintain.

When a Persistent Requirements Document Is Not Necessary

There are some instances when a requirements document or business solution is not necessary. Simply specifying what needs to be done is good enough. These include:

- The developer, business analyst, and customer work together consistently and in the same team.
- The system or modification is not persistent, such as a one-time-only report or query.
- Everyone involved can remember all of the requirements that comprise the solution document without writing them down in any manner.
- The circumstance is that no one will see anything but results; that is, no inspection or review of interim documents (requirements being considered an interim document).
- There is extremely little likelihood that there will be changes to the delivered product after implementation.

Obligation of Persistent Documentation

Once a document is prepared and submitted, especially after it has been approved, it becomes permanent. Because a document serves to freeze

communication at a specific point in time, the useful shelf life of a document is about that of a banana. To keep the document useful, the author or designated person must keep the document up to date. Failing to do so means that the time spent in initially preparing the document is wasted.

The more documents that are prepared to occupy permanent places on the shelf, the more documents the business analyst has to keep up to date. And the more documents there are that refer to the same problem or solution, the more care must be taken to make sure that all the documents refer to the same version. Considering the amount of documentation required in some organizations, it is no wonder a business analyst feels that his primary role is documenter.

To counteract this feeling that your world is made of paper, apply critical evaluation and a certain amount of skepticism to every demand for persistent documentation. Do as little of it as you can get away with. Focus on communication rather than documentation.

Ask of each documented item: "Has this been written down before? Do I need to write it down here? Is it necessary to keep this forever?" The rules of requirements specify that the solution document contain no redundancies. Documentation should never be redundant either.

You don't want to spend more time writing about solving the problem than in actually solving it.

Notice that in Figure 14.1 there is a clear separation between analysis and documenting, and that documenting comes only after analysis has been completed. This does not mean that you do not write anything down while analyzing—quite the contrary. When analysis is complete, you have a defined solution written down in some format that has been confirmed by the appropriate product stakeholders and perhaps accepted by the solution team.
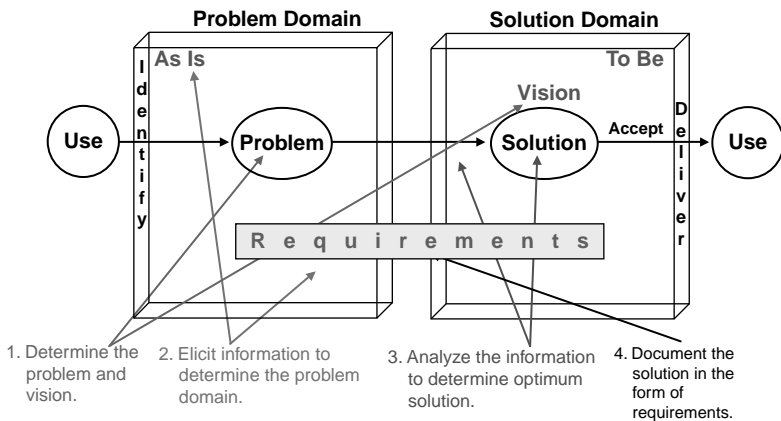
FIGURE 14.1 Documenting the Requirements

Now you refine those informal documents that define the solution into a formal solution document based on the standards of the organization or IT department.

The Anatomy of Requirements

"We don't get good requirements because we don't know what a good requirement is."

What Are Requirements?

Requirements appear in many forms. There are requirements for the project that guide the product implementation and deployment, technical requirements that define how the product will be built, and product requirements that specify what the product is and what it will do, among others. The business analyst is concerned with the latter set of requirements.

The problem and product may need to be defined at different levels of requirements to fully understand the entire problem and resulting solution. There may be a high level of abstraction that is understandable to management and business owners, a more specific level of detail that describes what the users or stakeholders need to do, and an even more exact set of requirements that relate the technical characteristics of the solution and how the system will support the business in solving the problem.

The product itself has different requirements that enable all parties to understand the various aspects of the product, such as requirements that specify what must be done for the product to function in a way that solves the problem, or requirements that specify the quality expected of the product by those who will use it to solve the problem.

The official IEEE definition of requirement is:

- A condition or capability needed by a user to solve a problem to achieve an objective.
- 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 documents.
- A documented representation of a condition or capability as in (1) or (2).[1]

Good and Valid Requirements

The requirements documentation process turns a set of good requirements into a set of valid requirements.

Good requirements are those that you and the stakeholders all agree completely and accurately solve the problem or some part of it. Good requirements do not have to be formal or structured or formatted. They just have to be written down. They may appear as bulleted lists, sketches of user interface screens on a whiteboard, workflow diagrams written on flip charts, collections of index cards containing user stories, notes on the back of an envelope, and so forth.

Valid requirements are the formalized set of requirements that define the solution to the business problem so that the business, upper-level management, and the solution team all understand it in the same way. The definition that we are using for "valid" is from *Webster's Revised Unabridged Dictionary:* "supported by facts or authority" and "capable of being justified, defended, or supported." The valid requirements meet a set of guidelines and rules, which help reduce ambiguity, keep the requirements focused on the problem and solution, and keep the requirements clear, precise, concise, testable, complete, correct, and traceable.

Let's walk through an example (see Table 14.2). Suppose the stakeholder makes the following request, "I want all my accounting reports printed out automatically on the first of each month." There are 25 accounting reports. Your first cut might be "The system may print accounting reports monthly." This statement is neither good nor valid. It does not reflect what the stakeholder requested, and it is ambiguous.

You revise the statement so that it does reflect what the stakeholder has requested: "The user may print 25 accounting reports on the first of every month." Your stakeholder confirms this requirement as stating what he has requested. He interprets "may" as "allow," as in "the system gives me the ability to print 25 accounting reports." This makes it a good requirement. However, the requirement as stated may not produce the result the stakeholder expected. The word "may" could be misconstrued from "allowing" to

TABLE 14.2 Comparison of Good and Valid Requirements

GOOD		VALID	
		No	Yes
	No	The user may print accounting reports monthly.	4.1 The system will print 31 reports on a specified day of the month.
	Yes	The user may print 25 accounting reports on the first of each month.	4.1 The system will print 25 reports as specified below on the first of every month.
			4.1.1 Report 1 will be . . .

"making a choice" as in, "The user may print, or the user may display . . . " or "The user may print 25 reports or 100 reports or 1 report" or "The reports may be printed on the first of every month, or not, or perhaps the 15th of every other month, . . . " The developer, to cover every possible option will create a comprehensive complex report generator allowing the user to select which type of report to print (accounting or other), how many reports to print, the format of the reports, the date and sequence of the reports, and even whether to print or display the report. The stakeholder does get the reports he asked for, but has to spend two hours each month setting up the report generator.

So you revise the requirement to be more valid. "4.1 The system will print 31 reports as specified below on the 31st of each month." Now the requirement is valid, but it is no longer good. The developer will produce exactly what is specified but the number of reports and the date to print are wrong.

The last revision of the requirement states, "4.1. On the first of every month, the system will print the following reports . . . "

Clearly you do not have to go through all these versions to get to the valid requirements. Typically we get good requirements as a result of our effective elicitation. The issue is making them valid. That starts with writing them at the right level of abstraction, which is discussed next.

Levels of Requirements

One of the common questions and points of confusion I encounter in my travels is in the varying definitions of the many categories of requirements. The levels of requirements show a progression from an abstract or general solution to a more detailed specification. The commonly accepted categories are business requirements, user requirements, and system requirements. The business analyst is typically responsible for the first two levels and many times is involved with the third. This does not mean that you have to create three physically separate documents.

Business Requirements Business requirements are the high-level statements of what is necessary to solve the problem and define capabilities the system or process must possess. Business requirements may also define conditions that must be met or constraints on the solution.

As shown in Figure 14.2, the source of business requirements is generally upper-level management or entities outside the organization, and each business requirement generates either user requirements or system requirements or both. Business requirements document the high-level business rules, capabilities, policies, and so forth that the business needs to do or to have done to exist as well as the high-level processes that align with the
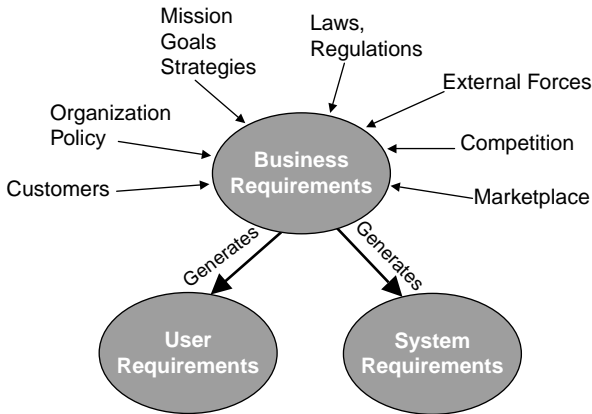
FIGURE 14.2 Business Requirements

business strategies to keep the business operational according to its mission. This includes:

- Organizational policies and procedures.
- Business rules.
- External influences such as laws, regulations, and organizational contracts.
- Competition processes and products.
- Organizational culture.
- Those interacting directly with the organization such as suppliers, customers, and vendors.
- The marketplace.

  Examples of business requirements taken from real-life requirements:

- The customer will have the ability to check out purchases without the assistance of a check-out clerk.
- Sales tax will be applied to all products except dairy and pharmacy.
- The new automated audit form will be used starting with fiscal 2003.
- All data entry screens will conform to the corporate user interface standards.
- Charges for uninsured subcontractors will be computed to the third decimal point and rounded up.
- Automated checkout lanes will allow payment by cash in addition to credit or debit card.

🔍 Example

Another fanciful example . . .

Management of the company noticed that its rate of repeat customer sales was down. It asked the business analysts to check out the situation. The business analysts traced some of the problem to the customer service department. The drop rate (the percentage of callers disconnecting while on hold) for customer service calls had increased by 26 percent over the past five months, which corresponded to the percentage drop in repeat customer sales. They further found that the average hold time (length of time a caller is on hold waiting for a connection) for customer service calls exceeded the industry average by over three minutes. Investigating further, the business analysts found that the average length of individual customer service calls had increased to a little over seven minutes. The business analysts were still investigating the source of the problem when the customer service manager decided that the evidence showed a direct correlation between the length of customer service calls and the reduction of repeat customer sales. The manager established a new rule in the department: No customer service call can last more than two minutes! He then asked IT to assist him in enforcing this rule.

This is a business requirement: all customer service calls will be completed in two minutes. Written out as a business rule, it reads:

> Once a customer service call is initiated, the system will terminate a still-active call in two minutes.

User or Stakeholder Requirements   This level of requirements is derived directly from what the process workers need to do to satisfy the business requirements. The workers require additional or different information or must have it presented in a different way. Workers have approaches to performing their jobs that may differ from the way the system or process currently operates. Their expectations of what the system should be doing and how it should be responding constitute their requirements of the system, whether right or wrong.

We record the users' preferences, predilections, and prejudices ("I want a pink screen of death") in the user requirements. We elicit how they do their jobs and look for the variances based on different users or different circumstances. The user needs are generally internal to our investigation.
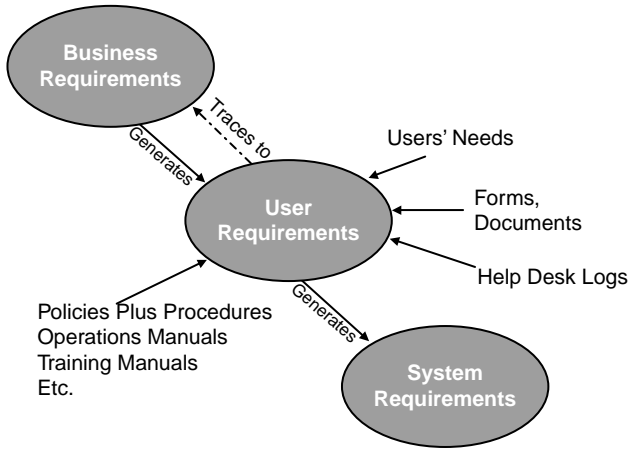
FIGURE 14.3 User Requirements

That is, they may never make it into the solution document. Not all user requests become requirements.

As shown in Figure 14.3, every user requirement is derived from and must be traced back to a business requirement. Every user requirement has at least one corresponding system requirement.

Here are some examples of user requirements:

- I need to see a warning message before I delete any project data to remind me what data I am deleting.
- We need to be able to enter up to 30 characters for a person's title, first and last name.
- The tabs on the screen should be in the following order . . .
- I'd like the error message to read " . . . "
- When the total button is pressed, the check-out screen displays the total of all items scanned.
- When the vendor number is entered, the system will display the vendor name and address.

In the capability maturity model integrated (CMMi), business and user requirements are collectively referred to as customer requirements. In many organizations they generally occupy a single volume, usually called the business requirements document (BRD), or to increase terminology confusion, a functional requirements specification (FRS), which contains sections for both functional and nonfunctional requirements.

🔍 Example

Continuing our fanciful example . . .

The business analysts, thwarted in their attempt to define the real problem, elicit information from the customer service representatives. During the investigation, the business analysts ask what the customer service representatives need to be able to comply with the manager's new requirement to end all calls within two minutes.

After a litany of complaints about the new policy, the customer service representatives relate that they have no idea how long calls take. What they would like to see is a clock on the screen that starts when a call is initiated and counts down from two minutes to zero.

Some prototyping sessions later, the business analysts refine the clock graphic to turn pink when there are 30 seconds left, and turn red and emit a soft buzz when 15 seconds are left.

These are user requirements—what the users need to satisfy the business requirements. The user requirements might read:

> When the call is initiated the system displays and activates a two-minute clock.

> When active, the two-minute clock displays time in digital format.

> When the call has been active for 90 seconds, the system turns the color of the clock to pink, and so on.

System Requirements The system requirements define what the system needs to do to satisfy the user or business requirements. For the most part, system requirements are transparent to the users and are rarely voiced by them. System requirements involve named databases, program interactions, hardware and software interfaces, network paths, and so forth.

As a general rule, system requirements are defined by the solution team and do not appear in the solution document created by the business analyst. System requirements generally appear in the technical specifications or design. However, there are instances where the business requirements may provide guidance in the definition of system requirements. For example, the users may request a specific response time or performance characteristic that will have an effect on the system requirements.

In some organizations, the business analyst creates both the business requirements, including the user requirements, and the system
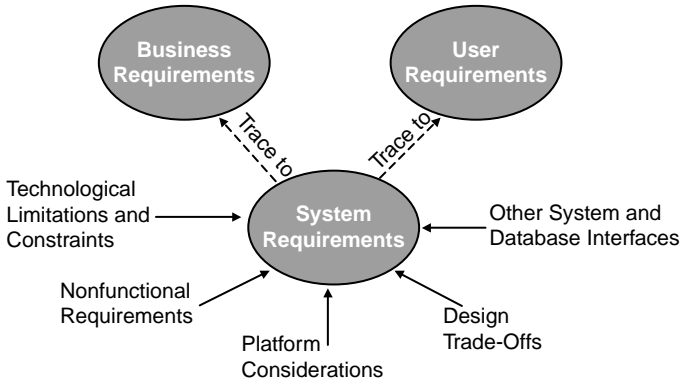
FIGURE 14.4 System Requirements

requirements. In a large U.S. bank, one group of business analysts defines the business requirements, and another separate group of business analysts defines the system requirements.

As shown in Figure 14.4, additional information is elicited which creates system requirements, such as design trade-offs and platform considerations. Every system requirement must trace back to either a business requirement or a user requirement.

System requirements define the internal system description. Examples of system requirements are:

- 95 percent of all terminal-initiated activities shall receive a response in 1.5 seconds or less.
- The system must be capable of processing 1,200 transactions in four seconds or less.
- The system must be capable of maintaining terminal response requirements while simultaneously processing up to eight background regions.
- Response time for worst-case latency will be less than 100 milliseconds.

A top-down structured approach in solution development starts with the high-level business requirements and works downward through the user or stakeholder requirements to the system requirements, which define the details for the developers. More agile and iterative approaches focus on a single function that may be a business requirement, a stakeholder requirement, a user story, or a use case. There is no concept of, or need for, levels of requirements. The business analyst or product owner, however, may use the levels as a way to organize or prioritize various functions of the overall solution. Looking at the requirements in terms of levels of abstraction also

---

🔍 Example
_____

The continuing fanciful example . . .

The customer service manager signs off on the requirements document and the business analysts turn the document over to the solution team. The systems analysts attack the problem. To put the clock on the screen for the customer service representatives, the systems analysts realize they need a way of synchronizing the initiation of the call with the clock and decide to use network time protocol (NTP) that requires NTP clients on each customer service representatives' station and an NTP server on the network. They also specify a faster refresh rate to display the customer service information on the screens to aid in completion of calls in two minutes. This requires upgrading the database management system and increasing the backbone network to 10 gigabytes.

These are all system requirements simply because it is unlikely a customer service representative would say, "Hey, while you are at it I would like to see network time protocol installed." Nor would we likely hear the customer service manager say, "Guys, we got to goose up our backbone to 10 gig to get enough throughput to offset the serialization delays at the routers."

The system requirements appearing in the system requirements specification might look like this:

There will be an NTP client on each customer service workstation.

Customer information used by customer service will be accessed via a cross-reference table.

---

helps us to remember that the product stakeholders and solution team naturally look at the same problem and solution in different levels of abstraction. Trying to force everyone to see the problem and solution the same way might cause a lot of confusion and frustration. It's easier to communicate when doing so at the appropriate level of abstraction for the audience.

Requirements Aspects

In addition to different levels, requirements also come in different flavors: functional and nonfunctional. At one time in the checkered history of requirements definition only functional requirements were necessary. Nonfunctional requirements were not a consideration. We did not ask the

product stakeholders about reliability, response time, security, availability, capacity, and so forth because we were defining applications that ran on computers (at that time there was no need for the mainframe distinction since there was just the computer). We could not make the functional application more secure than the computer or increase the response time for our one application, or make the application more reliable than the IBM 360 or the Univac 1180. However, once processing moved out of the safe confines of the computer and onto networks and then the Internet, we had to start worrying about such matters as portability, scalability, security, maintainability, and the like. Then when information systems moved out of the purview of a group of specialized, trained operators and into the mainstream of public use, more issues began to find their way into requirements documents: privacy, usability, auditability, globalization, and so forth. We were forced to distinguish between those requirements that defined what needed to be done—the functional requirements—and those requirements that defined the way in which it was done, namely the quality of the solution—the non-functional requirements.

Functional Requirements   Functional requirements capture the intended behavior of the system, what the user or process worker does with the system. This behavior may be expressed as services, tasks, behaviors, or functions the system must perform. Functional requirements are expressed as positive statements of action, written in the active voice.

Functionality can be defined with use cases or with user stories and are generally easier to describe and measure than nonfunctional requirements. This is because they are generally derived from the users' descriptions of what they do or need to do.

Nonfunctional Requirements   Nonfunctional requirements (NFR) capture the required properties or qualities of the system. They define how well some behavioral or structural aspect of the system should be accomplished. There are two types of nonfunctional requirements: those that are observable at runtime (for example, performance, security, reliability, user interfaces, and so on), and those that are not observable at runtime (such as extensibility, portability, reusability). IEEE standard 1233 (1998) lists the nonfunctional requirement categories. This list is in Appendix F.

Pay special attention to the definition of nonfunctional requirements because even when they are not explicitly stated, the expectation is that they will be met. It is difficult to elicit nonfunctional requirement information from the process workers and stakeholders (see the Example sidebar). The users are not likely to mention a particular area of concern, especially when they are involved with creating use cases of the system's functionality. Most of the nonfunctional requirements deal with backend or internal aspects,

such as privacy, security, and data integrity, which are characteristics the users generally do not think about. More likely, the users of a computer system assume that you know that response time has to be fast, the system should be available when they need it, that there should be enough capacity to store accessible data for a period of time, and so forth. However, leave one of those capabilities out—for example, response time—and the system, as a whole, will be considered a failure even when it performs required functionality precisely right.

Nonfunctional requirements can be very user-specific. One user's opinion of response time is that it is too slow, while another user working on the same process thinks the response time is a bit too fast. Think of the functional requirements as the meat and potatoes of the meal. The nonfunctional requirements are the sauces and condiments that add quality and individuality to the food. It takes extra time and effort to elicit information about the quality aspects of a computer system. Whether you delight the customer or just give them the functional necessities is what creates a quality solution.

Example

It is hard to define . . .

NFRs are harder for the process worker to define. I have had heard process workers say "I don't know, but I'll know it when I see it." One fellow, many years ago, sat at the keyboard to try the user interface for the first time and proclaimed, "It just does not feel right." No amount of probing could get a clear definition of what it was that did not feel right. I thought maybe we should put padding on the keys of the keyboard to make it feel better.

In one Ohio company, a business analyst described a software development project that delivered a functionally correct system. It did exactly what the users requested, and yet the users were not happy with it. User management claimed it lacked quality. One user group said they would not use it even if it did match the requirements; they finally signed off even though it did not feel right and took too much time. There were no specifications in the requirements that addressed these issues, and the response time and other performance measurements showed that the system worked at the same speed as the other systems the users dealt with. The users could not define what the issues were in unambiguous terms or why they felt it took too much time.

Forms of Solution Documentation

The format of the solution might be a formal document such as the business functional requirements (BFR), or the business area requirements (BAR), or the functional requirements document (FRD), or the business solution document (BSD) or any other prescribed format. However, the solution document may be:

- A deck of user stories on index cards.
- A set of use case models.
- A series of storyboards depicting the system flow.
- A set of wire diagrams showing what the Web pages are going to look like.
- A prototype of the software functionality, some words and sketches on a whiteboard.
- A requirements stack for use by a scrum team to define its sprints.
- Requirements as feature lists maintained on an electronic whiteboard.
- A set of loose-leaf notebooks, such that new requirements could be added to the notebooks when the changes were completed.
- A series of test cases.

Regardless of the medium and the message contained therein, the solution document must be understandable by both the business community and the solution team. Whatever the format, the requirements are written down and made accessible for review and discussion by all parties to the solution. The set of requirements which comprise the solution document, taken as a whole, constitutes the complete and accurate solution to the defined business problem.

To determine the right level of documentation to provide, use the solution team as the guideline. When the team has enough information to develop the solution, you have documented to the right level. When there are a lot of questions being asked by the solution team, it may be an indication that you need to add more information to the documented solution. When there are discussions and debate among the solution team over what you wrote, then you have not depicted the solution in as clear and unambiguous fashion as you thought. Make yourself available, agile-style, to the solution team throughout the implementation effort to clarify, explain, and augment the written word of the solution document. Remember that you have already gotten the solution confirmed by the business community. You are now producing a valid, persistent document for the solution team.

Write the Right Thing

When documenting the requirements there are some considerations to keep in mind. Here are a few of them.

Do No Harm

It is easy to solve a problem for a particular department of the organization while turning a blind eye to the impact that solution might have on other departments and individuals in the organization. After all, you are there to solve a problem for a problem owner. And the problem owner or department manager does not care about the other departments. When you are an internal business analyst whose primary goal is to increase the value of the organization this is not a tenable situation. Your efforts to increase the value in one area might be subverted by the loss of value elsewhere. Make sure there are no negative impacts outside the problem domain.

Finish the Analysis First

"I think our lead technical architect is a closet English major. He spends all the time correcting our punctuation and grammar. We can't get him to focus on what's important. Any ideas?"

Just as you do not want to come to conclusions until you have all the data, you do not want to formalize the requirements into the final document until the analysis is complete. Committing to paper too early gets you into edit mode, where concern is more for the format or language than for the overall content and the big picture of what is actually being said. Too many passes through the edit stage and the document, while being grammatically correct, no longer says what it was supposed to say. However, when the solution is defined in the end, make sure that it is grammatically correct, spell checked, and punctuated precisely so that the solution team (or anyone else) does not get stuck correcting your writing and not see what you are writing about.

Only Fill the Gap

"How can I streamline what I am writing in the requirements so that it is easier for them to read and assimilate?"

The solution should only define what must be done to solve the problem and close the gap identified during gap analysis. The solution document, in whatever form it takes is the last activity in the solution definition process. Creating the persistent solution document any earlier means that a great deal of time is going to be spent formally correcting, updating, adding new requirements, modifying the solution, and so forth, creating endless versions, which then must be initialed by the original approvers. This is not a game to play. Not only is this inconvenient to those approving the

document, it also diminishes the business analyst's image and credibility. There is no business analyst role that equates to documentation maintainer.

Write the Thing Right

"How can I write better requirements?"

> *"I write to better understand what I said."*
>
> —Philippe Krutchen

Think of the solution document as a model of the solution. Include diagrams, screenshots, pictures, as well as text descriptions. The solution team will use the diagrams more than the words, especially since they are most likely going to render the words into diagrams for development anyway. The business community may relate to a drawing of a screen layout better than to a three-page textual description of the same screen layout.

Tip

Important: Do not let documentation substitute for real communication. A solution document is only one of several techniques the business analyst uses to ensure that a consensus exists among all stakeholders.

Many business analysts get seduced by the necessity to deliver certain documents. Many are actually evaluated based on the documents they produce. They start making the document the primary outcome of their work. They believe that the document is also a way of perpetually assessing the author's abilities and productivity, so most of their focus and attention goes to perfecting the document in both content and form. Naturally the document contains the results of gathering information from the stakeholders. However, when the document is the end result of the business analyst's activities, more time is spent between the business analyst and the document than between the business analyst and the stakeholders. Ironically, focusing on the document tends to *reduce* the flow of information.

It is possible to implement a good system without documentation. It is not possible to create perfect documentation and still deliver the solution in a timely manner. The answer is to spend time communicating directly with the users and with the solution team until everyone understands what the solution is. Use the documentation simply to record the solution for posterity.

The Audience

The audience for the solution document you are preparing is the solution team. They are the ones who will be using it. Many business analysts assume that the audience is the product stakeholders because they are the ones who confirm and approve. And, yes, the user community needs to be able to understand the document so that they can confirm and approve, but consider this: After the document has been approved and implementation is under way, what do the stakeholders do with the document? The developers tear it apart, read it thoroughly, and use it as their guideline for development, testing, and so forth. Once the product stakeholders approve the document they have no further interest in it except perhaps to compare the results at the end.

Write the solution document with the developer in mind. The glossary, for example, should contain business terms rather than the technical ones we might be tempted to include for the business community's understanding.

🔍 Example

If you think that the ultimate audience for your solution document or requirements is the business community, consider the last time you wrote requirements for implementation. After a couple weeks or months, what is the solution team doing with the requirements (or the user stories, or the product backlog)? They are reading them, discussing them, analyzing them, tearing them apart and so forth. What are the product stakeholders who signed off doing with the document? Using it as a doorstop, piling it with a stack of reports from 2001, putting it under the shorter table leg to balance the table in the coffee room, and in general ignoring it. It's rare that the requirements even reappear in the business community at acceptance test time. They don't care if it matches the requirements, just that the delivered system solves their problem.

Valid Criteria

Valid requirements must meet, as much as possible, the following criteria:

- Unambiguous
- Complete
- Consistent

- Correct
- Feasible
- Traceable
- Measurable and testable
- Maintainable

Creating the solution document is the act of rendering the requirements into a format that complies with these criteria.

Well-Formed

Each form of the solution document has its own guidelines. For example, a use case description typically contains the following information:

- Use case identifier
- Actors (both primary and supporting)
- Preconditions
- Post-conditions
- Main success scenario
- Exception paths
- Alternate paths

When you are creating a formal solution document such as a business requirements document (BRD), functional requirements specification (FRS), system requirements specification (SRS), and the like, there is a formula for creating a well-formed functional requirement. Note that this does not apply to nonfunctional requirements.

A typical functional requirement contains the following structure:

- Condition (if, when, while, during, etc.)
- Subject
- Imperative (will, shall, must, etc.)
- Active verb
- Object
- Rule (optional)
- Outcome (optional)

A functional requirement might look like this:

When the vendor number is entered [condition], the system [subject] will [imperative] display [active verb] the vendor name and address on the screen [object] (see Figure 2.1) as long as the vendor is current [business rule] for visual verification [outcome] (note that the

last two phrases are somewhat of a stretch for purposes of demonstration).

Figure 2.1 in the requirements document contains the details of what the screen looks like.

The same approach is applied when using user stories. The user story typically has this format:

As a [type of user], I want [some particular feature] so that [some benefit is received].

The requirement above might be rendered as the following user story:

As an accounts payable voucher enterer [type of user], I want the vendor name and address displayed when I enter the vendor number [feature] so that I can visually verify I entered the right vendor number [benefit received].

Details of what the screen looks like and where the information is displayed are worked out between the developer and user.

In the end the important element in writing the solution document is that it is understandable the same way by everyone with an interest in reading it.

Canned Brains

"What is the balance between over-documentation and not enough?"

Even though you do not want to excessively maintain a solution document, it does need to be produced. My high school biology teacher, back in the early 1960s, was Gordon McKee. He had been teaching over 20 years when I took his course. He had a four-inch loose-leaf notebook of notes that was always on the lab table in front of him when he taught. He called it his canned brains. He probably could have taught the class from memory after all that time, and clearly knew the subject, yet he still had that notebook, and he actually referred to it from time to time during each day.

The solution document is much like that set of notes. Everyone may understand exactly what is to be done and verbally agree to it. You may have the specifications on the whiteboard and on index cards, making the formal documentation of the solution seem somewhat redundant. You may argue that the time is spent better writing new requirements than in documenting the ones you have already agreed on. However, there is a significant sense

of security when the decisions made last week are recorded and the brilliant ideas from a brainstorming session are inscribed in a form that can be revisited later. The solution document, regardless of format, is our canned brains, making sure that we are not sitting around in a meeting snapping our fingers in frustration trying to recall a part of our solution that was written on a long lost scrap of paper. As Peter Coffee says "It's easier to be smart tomorrow if you remember how you did it yesterday."

Requirements Ownership

"We are tasked with testing the results of the development efforts. We are not given much advance warning. Then when we use the requirements as a guideline to what we expect the system to do, it's all different. The technical team has made changes and we don't know what the system is supposed to do. How can we test it on behalf of the users if it isn't what the users asked for anymore?"

There was tension in the air of the conference room. I had to address a group of business analysts that I worked with for over a year. I had the basics of their process documented that identified changes to be made to their current way of doing things. They were not pleased with having to estimate the time they needed to elicit information and develop the requirements. They were not happy having to submit their acceptance test cases to quality assurance (QA) for review. Then I said the simple words, "The business analysts own the requirements." They actually cheered when I made this statement. All was forgiven. They were willing to go along with the other impositions. "Own the requirements" meant that no one, including project management, could make official changes to the requirements in the baseline except the business analyst who authored the requirements or another designated business analyst. Why were they ecstatic about this change, especially since it sounds like more work for the business analysts?

In the old process at this company, the business analysts developed the requirements and got them confirmed and approved by the business. After that, the solution team made changes to the system both legitimately and under the guise that, "The users will like this approach better," or, in other words, "We know what the users need better than they do." Project management also made changes in the best interests of the project. Project management, in this case, consisted of former technicians who had written the original system that was being maintained. The changes were never cycled back to the business analysts for review or even a FYI. When testing, QA used the requirements that the developers had used. All was good. However, when the users reviewed the results they saw something different than they had specified and approved and they were not pleased. When they

asked the business analysts about the unauthorized changes, the business analysts were broadsided by the complaints, since they were totally unaware of the alterations.

In the new process, the developers and project management can still make changes. The changes cannot be recorded into the requirements by anyone but the business analysts. Therefore, when it comes to testing, QA creates test cases from the baseline requirements. This process forces the solution team to include the business analysts in on the changes, regardless of what the changes are, whenever the changes affect the customer-approved requirements. The business analysts, in turn, notify the business when any changes affect them.

This is the way it should be—the business analyst owns the requirements.

Complete the Process

The process is completed with a solution document that has been approved by whoever needs to approve it.

Get the process workers and system users—the ones who will actually be executing the new process or functions—to confirm that the solution document, or their part of the solution document, will work just fine. Then get the approval from those with the authority to approve. The approval, in whatever form it takes, is an official blessing by the organization for the project to continue and for the development team to produce and implement the solution. This approval cannot occur until the solution has been confirmed.

In general, after the solution is confirmed and validated, it is approved with at least three signatures: the business analyst who authored the document, someone in the business unit (usually the problem owner or the executive decision maker), and the person on the solution team to whom the document is delivered. The reasons for the acceptance signatures are:

- Business analyst: This is indeed my work and I stand behind it.
- Business owner: I concur that this set of requirements constitutes a complete and accurate solution to my problem.
- Solution team recipient: I understand and accept this document and can implement the solution from it.

In more agile approaches, the approval for the work to be done in the next iteration comes from the product owner or authorized representative.

Once approved the product moves into the implementation phase and the solution team produces the product. This is not a relay race, and the

business analyst is not simply passing the baton off to the solution team and retiring from the race to stand on the sidelines and cheer. This is the point that the business analyst has turned over the diagnosis of the business problem and the solution to the specialists to implement the cure. The business analyst, much like the internist, has to be involved with the treatments to make sure the diagnosis was correct and there are no adverse reactions along the way. And the business analyst, like the internist, needs to be present when the specialists come up with a better treatment plan, so that he can take the notice of change back to the patient, or stakeholder. Exactly what does the business analyst do while the solution team is engaged in the implementation of the solution? We explore the activities of the business analyst during solution implementation in Part Five.

Note

1. IEEE Standard 610.12, 1990.

Producing the Product

Every project needs an individual champion . . . to advocate for the product.

—Dean Leffingwell

Implementing the solution is the final stage in solving the business problem. During this time the business analyst may find their direct involvement in the project is somewhat reduced while the solution team carries out the coding, testing, database developing, architecting, building, purchasing, and other activities.

During the implementation the business analyst continues to perform the roles of facilitator, communicator, and educator to ensure that the goal of solving the problem is not lost while solving it.

Product Champion

The concept of product champion comes from marketing and research and development. The product champion brings organizational resources to bear on both the definition and the development of a product. The product champion drives the product into existence.

Here is a profile of the successful product champion.[1]

- Knows the product stakeholders—all of them,
- Is agnostic and objective about all the problems and issues in the business community.

- Has business experience in the domain.
- Can speak intelligently and confidently about the issues.
- Is a good facilitator.
- Works and plays well with others on all sides of the issues.

What is the function of a product champion?

- Accepts responsibility that the product delivered to the business solves the business problem.
- Enhances the solution team's ability to produce the product.
- Defends the business community's need to have the product.
- Represents the best interests of the customer(s) and the product, steering product development in the right direction.
- Balances the needs of the solution team against the needs of the product.

This description of the product champion sounds like the description of the business analyst we have come to know and love.

"The product champion is the one person who is officially responsible for delivering the product. This person helps the stakeholders and project manager reach a shared vision for a product, and then defines and initiates the product within that vision."[2]

By adopting the role of product champion, not only will you hold on to the vision of the product and solution so that the solution team does not go astray and build something that is different than expected, you will also resist attempts by the business community to change the product along the way to include additional features that are not part of the solution.

Eyes on the Prize

It is an easy thing to lose sight of the problem. IT projects sometimes go on for months and years. There is a lot involved and a lot at stake. Larger, more mission-critical projects are also more political because they affect more of the organization. As a result there may be compromises and diversions and, occasionally, outright sabotage. In addition, there are the normal distractions that occur in any long-term project: diversion of resources to other temporarily higher priority or more immediate problems; changing personnel on both the project team and the business management teams with the new players having their own vision of the product; changes in the business and the marketplace, and so forth. Each of these distractions and diversions has the effect of potentially altering the project from its intended goal of solving the stated and approved business problem. The project manager,

who is likely more in tune with the politics, will change the direction of the project when so directed as long as the deadline and/or budget are adjusted accordingly. The business analyst, however, must keep focus on the problem that is being solved, and the product that will solve it, even when it appears no one else is doing so.

The business analyst creates a strong and compelling vision of the solution and can ask the right questions of the solution team to gauge whether the project is on track—not on track with budget and schedule, but on track toward the vision that will provide the solution to the business problem. The solution is a collaborative effort of all parties throughout the solution life cycle. The business analyst is the catalyst for the collaboration: acting as the customer-facing member of the solution team and representing the business in the various implementation decisions to be made. The business analyst moves from center stage (defining the problem and the solution) to a supporting role (clarifying, verifying, confirming, and assisting); however, the activities of the business analyst are still integral to the successful solution.

Monitor the Product

Software development is not a solo, intellectual task. Rather, it is a collaborative, social task that requires lots of communication.
—Peter McBreen

Once the solution document has been approved and/or accepted, the business analyst's role varies during the actual development process, based on the method of software development in use by the solution team. The business analyst has a role similar to the architect during the construction of their design. The architect observes the construction and makes sure that it is still following the architectural specifications. When the builder needs to change the plans for technological reasons, the architect changes the specifications and makes sure the stakeholders (owner, city and county government building inspectors, etc.) approve. Similarly, the business analyst changes the solution document to reflect valid design changes by the solution team.

"Where does the business analyst fit into our software development life cycle?"

Solution development is the process of turning the solution definition prepared by the business analyst into an operational system or process.

There is a tendency to believe that the business analyst's job ends with an approved set of requirements. And perhaps at one time those who defined requirements deposited them at the doorstep of the project team and called it a day. The true business analyst, however, continues to work with the solution team throughout product implementation.

During development, you play the role of customer representative on the solution team. You work with the solution team to make sure that the results of the development effort still solve the business problem.

During design, the business analyst assists the systems analyst to make trade-off decisions among various technical solutions, bringing any impacts to the solution document back to the business for review. During the build and early testing phases, the business analyst makes sure the solution document still matches the solution. During late testing stages—system and acceptance—the business analyst prepares test cases for users, and participates fully in the acceptance test stage. After the product is delivered, the business analyst makes sure the solution is successfully transitioned into the business environment.

Entering the Solution Domain

Each organization has its own set of entry criteria that moves a project from the problem to the solution domain. Usually the trigger is the approval signature on the solution document. However, there are a number of conditions the business analyst wants to see met to make sure the product is ready for development:

- The business process in which the problem exists has been completely and accurately defined and confirmed by the business.
- The conditions that cause the problem to exist have been identified.
- Solutions are being offered to eliminate or ameliorate the conditions.
- One solution has been approved by the business
- The solution team accepts the task of producing the solution based on the solution document.

Once the the solution team agrees to implement the solution document, as a 100-page document or a user-story card, the process is in their hands until the solution is ready for acceptance testing.

Development Processes

As business analyst, you might prefer a linear process of development, so you can gauge what the progress is and facilitate the correct feedback from the product stakeholders. You might like to know just when the design is complete and programming has started, when the product is ready for each of the stages of testing, and so forth. Unfortunately, it does not happen that way. The development process generally is iterative and incremental whether or not it is planned that way.

The project manager and solution development team have a wide range of choices for the way they want to develop and implement the software. The following is a quick overview of the software development approaches currently in use.

Linear Development

Linear development, commonly implemented as the waterfall, has a single delivery of the product at the end of the project. Each phase of development is executed once. In this approach, the product scope and solution document are clearly defined as completely as possible at the beginning of the project. Changes to the solution document are controlled through a defined change-management process.

Most current development processes are somewhat linear in nature. Many of the products delivered are small enough to be completed in one iteration and are not subdivided into increments. There is also a perception of control with the linear, structured approaches that gives upper-level management more comfort. Upper-level management has a need to know where things are in the development of a solution and when things are going to happen so that they can plan resources and funding accordingly.

In this linear model, the business analyst's involvement follows a curve as in Figure 15.1. The maximum involvement in the solution is at the beginning and end of software engineering.
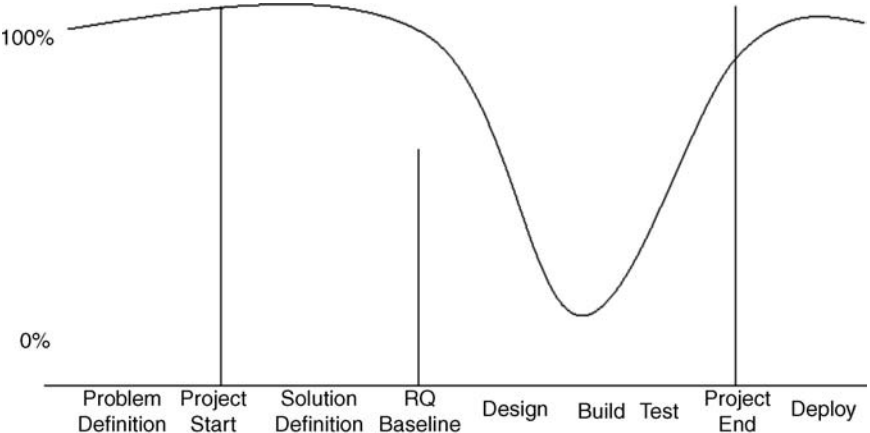
FIGURE 15.1 Involvement of Business Analyst in Linear Development

''[Name of department head] is having discussions with the project team about what to deliver since they can't deliver the whole thing. Is this something that I have to get involved in?''

Incremental Delivery

Incremental delivery produces some functionality in smaller discrete releases. The business community and the development team agree to divide the totality of the solution into deliverable, operational segments that can be placed into production so the users can gain benefit out of some of the new software and system while the development team is continuing work on the rest. This is like building a house in such a way that you can live in the part that is finished while the builders work on the next section.

The business analyst is deeply involved in an incremental delivery approach. The delivered increments should be based on business, rather than technical, rationale. The increments typically are aligned with the functional goals established by the business analyst during product scope definition.

In an incremental delivery approach the business analyst assists the business and project managers in determining the increments that make sense. Each increment should deliver something that provides value to the business community regardless if the increment goes into production. The business analyst provides:

- Definition of the level of quality for the release.
- Time frame that the business can accommodate.
- How much change can be assimilated by the business at one time.
- The definition of how the business will know that all changes are effective.
- Roll back positions based on deadline demands from business.

''With iterative approaches, such as the unified process, is it still necessary and wise to have a single scope statement at the beginning of the project?''

Iterative Development

The iterative development approach provides an opportunity for all to see what the overall solution will look like early in the development cycle and come to agreement that the solution is acceptable and correct. At the end of each iteration, after feedback has been obtained, the requirements can be further detailed in the solution document and passed on to the solution

team in a phased linear approach, or delivered as user stories, use cases, or backlog for use in a more agile approach.

The business analyst is responsible for getting the feedback from the business community after each iteration and for keeping the requirements current amidst the continuous changes that are a hallmark of iterative development. You determine the results of each iteration (what the business sees), orchestrate the review sessions, and record and act on the feedback.

"We're using agile development (XP), what is my role as business analyst in this situation?"

Agile Development

Agile software development is an approach epitomized by Extreme Programming (XP) and Scrum, which combines the incremental and iterative approach with timeboxing and specific development practices. In agile there is a short, fixed timeframe specified and the solution team determines how much software can be produced in that amount of time.

Scrum defines a product owner as the representative for the business community. The description of the product owner sounds very much like our description of the business analyst role.

Dr. Steven Gordon suggests that it could make a big difference when you plan the agile iterations so that the business analyst spends each short iteration (one to two weeks) simultaneously:

- Working with customers and SMEs to fully understand the requirements for the next iteration.
- Standing-in as the proxy customer for the developers to produce the current requirements.
- Getting feedback from users on the software delivered from the previous iteration.

Implementing the Solution

While the solution is being implemented the business analyst's primary activities are in facilitation, mediation, and negotiation. You play the role of enterprise communicator, making sure that all changes are noted, evaluated, and reviewed.

During systems analysis and design, the designers are specifying how the solution described by the business analyst will actually be put into production. They will determine the combination of software, hardware, networks, people, and data necessary to make the product actually work for
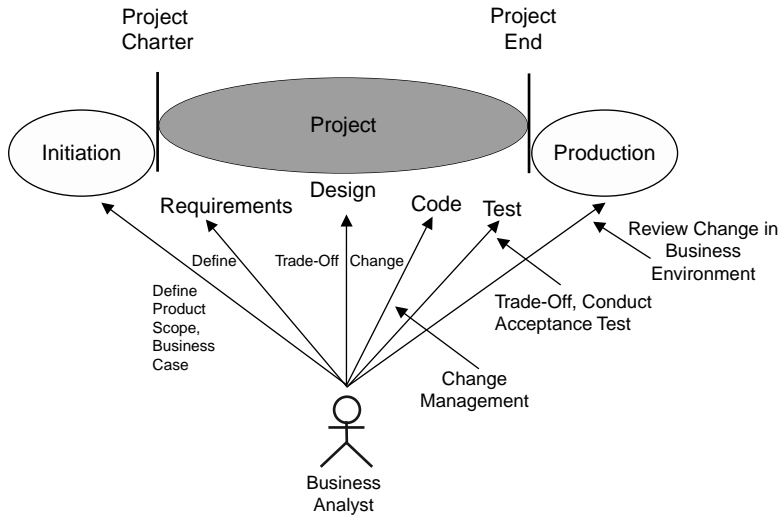
FIGURE 15.2 Business Analyst in the Development Life Cycle

the users. The design that is produced by this phase must accurately and completely reflect the solution as it is defined at the time.

As shown in Figure 15.2, the business analyst is involved throughout the project and software development life cycles. We've already discussed the activities of the business analyst during the initiation and the early stages of the project when the problem and solution are being defined. Once the solution is accepted and moves into the implementation stages, the business analyst does not leave the document on the solution team's doorstep and run off giggling into the night.

The design phase consists of making trade-offs, mostly technical in nature, about how the solution will be accomplished. The solution document specifies only *what* must be done to solve the problem. The design provides the specifics of *how* it will be accomplished. The designers may come up with elegant, efficient solutions that do not fall within limitations imposed by the business solution. On the other hand, their approaches may use technologies not known to the business analyst or customer at the time and may present a better solution than the one the business analyst conceived.

An example: The business analyst has an assignment to automate the last of a series of government forms. The problem is that some forms are produced using keyboard entry, and some are done manually. The manual forms take too much time to produce and have too many errors. The organization envisions having all the forms produced in the same manner. When the business analysts turned the requirements defining their solution over to

design, one of the designers suggested using a scanner to input the source material and have it translated to the output format instead of keying in all the information. The business analysts took the suggestion back to the customer and it was approved. The customer then elected to eventually move all forms to the scanning solution.

In most cases, however, the trade-offs are made within the solution space provided to the designer and do not affect the business solution.

Keep the Light on

Although more common with a waterfall or linear approach to software development, in IT the project team is accustomed to "going dark"—removing itself from all contact with the users. When the requirements have been signed and baselined, the development team stops all conversation and communication with the business community. Part of the reason for this is to allow developers the freedom from changes to the specifications that interrupt the flow necessary to software creation. They feel this freedom is necessary to develop the solution in the time allocated. Sometimes the development team retires to another location and works in what is called skunk works. One could assume that once completed, the users and other stakeholders are less likely to make changes and will want the product in production; therefore, they accept what was developed without change, holding the changes off until the next release, or never.

Going dark may also apply to the business community as well. During the time that it takes to actually produce a solution to the business problem, those that are affected by the problem may distance themselves from the solution. They may get used to working around the problem. The longer it takes to solve the problem the more likely the impetus for change will lessen, until those who initially lodged the complaint may decide that the change is no longer necessary.

Regardless of where you, the user community, and the solution team are located, encourage the flow of information among everyone during development so that the business community is always aware that the change is coming, and also to let them know when the solution team changes things. Remember at the end, when the product is delivered, there should be no surprises.

Things Change

During solution development there will be changes. The changes come from the business and from the solution team. Sometimes there are more

changes from the solution team than from the business during solution development. As business analyst, you are part of the evaluation of every change request regardless of the source. You review the request and confer with the source of the request or the problem owner (depending on the politics). Your primary concern is to get the answer to the question: Does this change help achieve a solution to the problem?

This is basically a filtering activity (Chapter 6). You are reducing the number of change requests the project manager (or change control board (CCB)) has to review by eliminating those change requests that are examples of the tummy-tuck syndrome discussed in Chapter 7. Some submitted changes are legitimate and reflect new information obtained after approval of the solution document. Some might be the result of changes in the business environment.

At the same time you are helping the stakeholder clearly define the benefit of the change to the product. A large percentage of implemented software is never used by those requesting it, so some up-front critical questions about the utility of the request are in order.

When you have confirmed that the proposed change meets the criteria of being part of the solution, forward the change to the project manager and/or CCB. The project manager evaluates the change by asking two questions:

1. Can this change be done within the current schedule without adversely affecting the quality of delivery?
2. Can this change be done within the current budget without adversely affecting the quality of delivery?

When the answer to both is yes, the change can be incorporated into the solution document. Otherwise, there must be negotiation for more time and/or more resources. The business analyst may be called upon to conduct or participate in those negotiations.

Checkpoint Charley

A business analyst manager had this to say: "There is significant communication up and down the chain in which the business analysts are not involved. It seems like this with every project. We prepare a business requirements document reflecting what the users want and then the developers do their own thing, disregarding our document. And they don't tell us. They are all discussing it but once they get the document, we're out of the picture. And the business expects us to deliver what they saw and approved in our document."

A resolution to this problem is a formal mandatory meeting held when the design is complete: the third of the three checkpoints, called Checkpoint Charley. The goal of the Checkpoint Charley meeting is to make sure the business analyst understands the implementation approach being taken and agrees that the approach solves the business problem. The meeting is also for the business analyst to capture the changed or new requirements that have resulted from the detailed technical review of the solution document while the analyst was not in contact with the development team. This is a good way to make sure nothing slips through the cracks.

The technical lead on the solution team presents the technical design to explain how the design solves the problem. In attendance are all parties with input to or requiring information from the technical design, including other technical team members, business analysts working on related projects or work requests, and any other member of the project team. Management is not typically included in the Checkpoint Charley meeting.

The business analyst corrects all requirements in the solution document with the valid changes identified by the solution team. Should a change make a significant difference to the solution, the business analyst presents the changes to the business for approval.

Again, in agile development methods a Checkpoint Charley is not necessary. The developers and product owner or customer are in constant communication so changes or issues that come up during development are exposed and discussed as they occur. Even so, the product backlog has to be groomed and reprioritized and some description of what is going to be delivered has to be maintained for such incidentals as user manuals, help desk guides, and online help information.

The Watchdog

"How can we make sure there are no surprises at the end when we are delivering the solution?"

The overall responsibility of the business analyst during the solution development period is to ensure that the developing solution still solves the business problem as it was defined by the requirements. This means continual negotiation with the development team and the business as implementation details change the original concept and requirements.

Once embarked on solving the problem, the solution takes on a life of its own. The overall goal of solving the problem can get lost in the flow of iterative interim solutions.

A business analyst once told me: "In an ideal world, the developers would develop what the business requests; however, in my world, some

developers do as much or as little of a scenario as they feel like doing. Sometimes it is because they don't want to put in the effort and sometimes it is because they don't have the knowledge to do it the way the business requested. Or, they want to show a cool new feature they can add quickly. In these cases, I, as the business analyst, need to keep the developers on track with the business goal. The businesspeople are busy running the business during the iteration so they need a spokesperson to keep things progressing with their needs in mind."

When there is a divergence between what is being developed and the solution statement, there are two courses of action:

1. Reject a potential function or feature as not contributing to the solution of the problem.
2. Revise the solution statement to include the feature or function.

In either case, it is your responsibility to make sure the specified, agreed-on solution matches the product resulting from the development effort.

As Figure 15.3 shows, during the development process there are changes to the business requirements. The overall solution may not change. The system analysts, programmers, testers, database administrators, and everyone involved are going to suggest ways to implement the solution that cause alterations to the requirements in the approved business solution document. Many of the changes are minor. Some require acceptance by the business community before being adopted into the design. Some may cause the solution to be less than satisfactory and be rejected. The first checkpoint on these potential changes is the business analyst who confirms that the change does not negatively affect the solution; the next checkpoint is the project manager who will evaluate whether the proposed change is viable within the timeframe and resource limitations of the project.
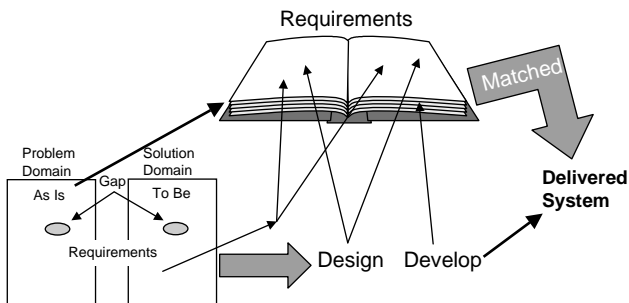
FIGURE 15.3 Keeping the Requirements Up-to-Date

The Essence

During the development effort, when product stakeholders have gone back to their day jobs and are learning to live with the problem until solved, and developers are slaving away over hot keyboards pushing and pulling Java instructions and tickling the database tables, the business analyst plays the role of product champion. The product is described in the solution document, so the business analyst focuses on making sure that the solution document description of the product matches what is actually produced, and vice versa.

At some point toward the end of product development the solution team has completed the creation or modification of the software and has tested it to be sure it does what it is supposed to. The solution team turns the product over to the user community, the quality assurance department, or the business analysts to undergo final testing. Regardless, of who manages the acceptance testing stage, the business analyst is formally or informally involved. Proving that the developed product actually solves the problem in a quality manner is the next step in the business analyst solution process. We discuss the role the business analyst plays in assuring the quality of the solution delivered to the business in Chapter 16.

Notes

1. Linda Rising, "The Product Champion," *STQE* 5 (3) (May/June 2005): 44–48.
2. Ibid.

Confirm the Business Problem Has Been Solved

In testing, instead of thinking pass versus fail, consider thinking problem versus no problem.

—Cem Kaner

Business analysts are not professional testers although many spend a lot of their time doing nothing but testing. Testing, or quality control, is a specialized discipline with its own associations, organizations and certifications. Testers play by their own rules and have their own practices and procedures. Phrases like test driven development (TDD), exploratory testing, test scripts, test frameworks, test harnesses, white box and black box testing, and the like constitute a different lexicon and a different world from that of the business analyst.

Despite that, the business analyst has a strong relationship with the software testers even if the business analyst never meets a single tester. The goal of testing is to prove the product has achieved an acceptable level of confidence that it will behave as expected under all circumstances of interest. To accomplish this goal, the testers need a source of correct behavior, the specification of the circumstances of interest, and what constitutes an acceptable level of confidence. The business analyst supplies that information. There are many levels of confirmation for which the business analyst is responsible, including:

- Establishing the basic criteria against which quality assurance judges the quality of the product.

- Confirming the final definition of correct behavior—the solution document—matches the product being delivered.
- Making sure the product solves the original problem.
- Proving the solution is effective and stays effective by measuring the results.

Correct Behavior

"What is the connection between requirements and testing?"

Correct behavior is the documented description of what the system is supposed to do in reaction to various events such as the arrival of information, the pressing of a submit key, and so forth. It establishes the baseline against which results of tests are validated. When you describe the problem domain and specify the solution document you are defining what correct behavior is.

Recall our discussion about defining the problem domain first (Chapter 12). The definition of correct behavior can be found in the:

- Definition of the problem domain—the "as is" that, once defined, constitutes the system baseline.
- Definition of the changes to the problem domain necessary to solve the problem.
- Definition of the system in the system design.
- Benchmarks from external sources.
- Policies and procedures of the organization.

Note that with the exception of system design, the business analyst supplies all the definitions of correct behavior.

Acceptable Level of Confidence

There is no reasonable way to test every conceivable condition to ensure the software is totally free of defects. The cost of doing so is prohibitive. For example, to achieve one hundred percent confidence that all combinations of a string of ten uppercase alphabetic characters work in a given program, there are 1.4×10^{14} combinations to test, which would take 4,500 years at one millisecond per test to complete.

Cem Kaner introduced the term *good enough software*. Good enough software solves the problem and achieves a level of confidence in the

system that the business community is comfortable with. The business analyst establishes the business' level of confidence based on the level of functionality the business needs to solve the problem.

To do this, determine the acceptable level of confidence in agreement with the problem owner or other stakeholders. Establish both acceptance criteria and minimum acceptance levels. When you ask, "What is it going to take for you to believe we have solved your problem?" the answer is the acceptable level of confidence. And you have done that back in Checkpoint Alpha (chapter 8).

Circumstances of Interest

The circumstances of interest circumscribe technical and business areas in which testing will occur. For example, when the organization standard for a desktop platform is Microsoft, there is no need to test on any other platform. Other platforms are not within the circumstances of interest. Similarly, when the organization has 1,200 employees and is growing at 10 percent a year, it is not of interest to check whether the system will handle 10,000 employees.

When you determine the limitations and constraints of the final solution (chapter 9) by defining the problem domain, product scope, and business constraints, you are specifying the circumstances of interest.

"The quality of the system depends on the success of the testing; the success of the testing depends on the quality of the requirements."

In theory, you can and should legitimately only test what has been stated in the requirements. Of course, in theory, the requirements are defined completely and correctly and there need be no further communication between the users and testers. In real life, testers routinely go back to the users and ask for clarification or suggest new ideas and approaches, which turn into new requirements. For example, when there are no requirements specified for reliability, and no reliability standards exist for the organization, the testers have no correct behavior to test against. The testers will either check with the business to determine the requirements for reliability, or not perform reliability testing at all. The best option, outside of an agile approach, is for the business analyst to define all the functional and nonfunctional requirements in advance. That way the developers and testers know what the reliability expectations for the system are, or they know that no one really cares. Your solution document in whatever form it is rendered in should at some

time contain every applicable requirement even by reference ("The sys-
tem will adhere to the current organizational security policy") so the
testers test all facets of the system.

The Testing Game

There are four basic stages of test execution that software goes through from
the level of code to finished product. A different group in the solution team
is responsible for each of the stages. The stages are:

1. Unit testing.
2. Integration testing.
3. System testing.
4. Acceptance testing.

The business analyst's involvement in the physical testing of software
increases with each stage.

Figure 16.1 shows the stages of testing. The solution team performs the
right-hand testing stages (unit, integration, and system) to prove they built
the product according to the development process in use by the team; the
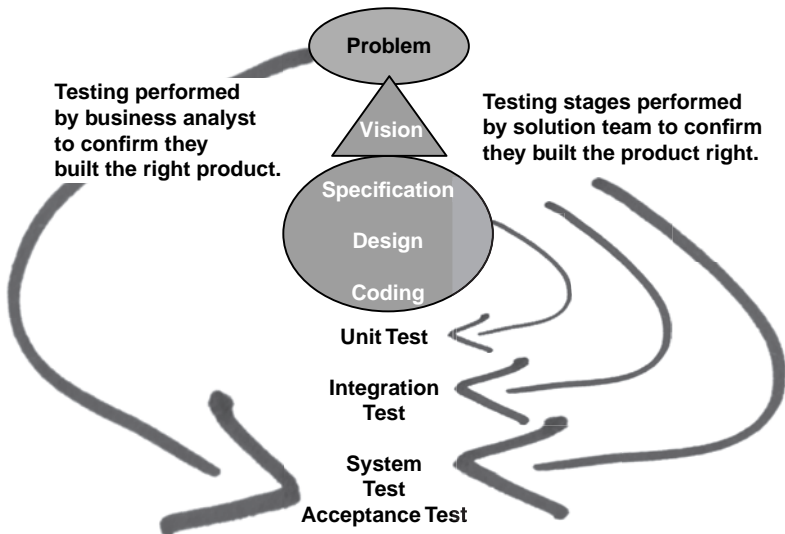
FIGURE 16.1 Testing the Solution

unit test confirms the coding is correct; the integration test validates the system design; and the system test confirms the specifications. The assumption is that when the process is correct, the product is also correct. The business analyst and business community, specifically the users, under the direction of quality assurance where applicable, execute the last test stage to prove that the right product was built and the problem is solved.

Unit Testing

Developers perform unit testing. The developer tests the software program to make sure the program performs as specified. A unit is the smallest piece of testable code. As such, unit testing is generally aimed at evaluating the code as implemented, rather than evaluating conformance to some set of requirements. The basic purpose of the unit test is to confirm to the developers that the individual code they have written to solve part of the problem works as designed.

The business analyst is rarely involved with the unit-testing phase.

Integration Testing

Developers also typically perform integration testing, either through an automated integration test mechanism or by aggregating unit-tested code into components and testing the components together. During integration testing the functional software is tested against databases, external interfaces, and so forth. The concept behind integration testing is simple. After each of the individual units are tested and work successfully, the units are put together in a preplanned order so they can work according to the system design. Integration testing is primarily about proving that the system design, which defines how the solution will be implemented, works.

In some companies, the business analyst gets involved with integration testing, primarily as a voice of the business community to respond to functionality questions and keep the communication open between users and developers.

System Testing

An independent test group usually does system testing. Some organizations combine the integration and system-testing stages into a functional-testing stage. The system-test stage, according to the IEEE 610.12 (1990) standard, is "testing conducted on a completed, integrated system to evaluate the system's compliance with its specified requirements." In other words, system testing is about making sure all requirements, functional and nonfunctional, have been tested. System testing is not one set of tests. It is a combination of

tests that verify a number of different characteristics of the solution, such as security, response time, reliability, functionality, capacity, accessibility, and so forth. Each characteristic may require a different set of tests and even a different test environment. The general purpose of the system test stage is to make sure the assembled components function together as a whole, in a quality manner according to the requirements, and that the product is ready for acceptance testing by the user community.

The business analyst is more involved with system testing because the correct behavior that the system testers are using to validate the tests is the solution document prepared by the business analyst. During this stage the business analyst may be called upon to negotiate requirements. Many times requirements are overly ambitious and cannot be achieved in the solution development time frame. For example, a requirement might specify a certain response time that testing shows is unachievable. You may be in a position of negotiating a new response time characteristic with the business community so the product can be delivered by the due date. Additionally, the business analyst may step between the developers and testers if their interpretation of the requirements differs such that the testers are declaring a defect, and the developers are refuting it.

Another role you may find yourself playing during system and acceptance testing is that of mediator. The testers identify aspects of the product that they do not think is correct. The developers believe they have created the system according to specifications. There is some disagreement as to the definition of a defect according to the requirements. In steps the business analyst to make a final determination, sometimes after consulting with the stakeholders, about the disposition of a defect.

🔍 Example

At a Fortune 50 company the quality assurance testers were incentivized based on the number of defects they found. The developers were incentivized based on a low number of defects delivered. Needless to say, there was a constant conflict about whether something was in fact a defect. To help reduce the constant conflict, the business analysts, who defined the requirements on which the definition of defect was supposedly based, were brought in as mediators to determine the outcome of the defect. Fortunately policy was eventually changed, removing the defect count as a measure of productivity and quality.

"How do we make sure user-acceptance testing is done correctly? What is the business analyst's role?"

Acceptance Testing

The acceptance-test stage confirms that the system under test solves the problem. The acceptance test generally involves the business community although not always. The test cases and scenarios are written in business and results-oriented terms, rather than in terms of specific inputs and outputs.

The acceptance tests are a set of test scripts, scenarios, cases, and procedures that prove the acceptance criteria have been achieved. The business analyst starts creating and assembling these tests as soon as the acceptance criteria have been defined.

The acceptance test cases, scenarios, and procedures, once successfully executed, provide a complete regression test suite for the changes that have been made to the system. The acceptance test suite should match the solution document, which in turn should match the delivered system.

Manual procedures that have changed in conjunction with the new system being implemented should also be tested. This usually is done by verification or observation, generally after the implementation.

The business analyst's goal in acceptance testing is to ensure that the implemented product does, in fact, solve the problem and solve it in a quality manner as prescribed by the very requirements you defined. Various organizations have defined different roles for the business analyst to play in acceptance testing:

- The business analyst defines acceptance test cases and/or scenarios and executes acceptance tests on behalf of the user community, including identifying defects.
- The business analyst defines acceptance test cases and/or scenarios and users execute the acceptance tests, including identifying defects.
- The business analyst defines acceptance test cases and/or scenarios and manages the acceptance test stage while users execute the acceptance tests and identify defects.
- The business analyst creates acceptance test cases and scenarios and turns them over to quality assurance which manages the acceptance test stage.
- The business analyst assists the quality assurance department in defining acceptance test cases and/or scenarios and advises during the acceptance test stage.
- Users and the business analyst define acceptance test cases and/or scenarios; the business analyst then executes the tests and users review results to identify defects.

- The business analyst defines acceptance tests and/or scenarios and test execution is done by an outside organization, with the business analyst reviewing results and identifying defects.
- The business analyst acts as a resource for users who perform testing on their own and turn the results over to QA, management, or the business analyst.

"How do you know that the system you are testing really is ready for deployment?"

There are three elements that the business analyst uses to confirm that the problem is solved:

1. The vision.
2. The acceptance criteria as described by the problem owner: what the business needs to see to agree that the problem has been solved.
3. The set of tests that the business analyst devises that create the proof. When tests are executed and successfully return positive results, the business can see that the acceptance criteria have been met and the problem solved. The business analyst performs the tests first to satisfy himself that the problem has been solved.

An important consideration when defining acceptance tests is the involvement of the user. You want as much user input to the acceptance tests as possible. Even when the user is not involved in the actual testing, the effort is still done on behalf of the user. Independently creating the proof that the solution does what it is supposed to do, absent the user, is asking for problems later on. The users may not be able to go off on their own and create valid tests. The business analyst, however, can use input from the user to fashion the scripts for testing. The simple question to ask users or SMEs about their particular area of concern or the area about which they are the most concerned is: "How will you know the new system does what you want it to?"

Should that question not generate a usable answer, you might try these two questions:

1. "What is the first function you are going to try when the new system is implemented?"
2. "What do you expect to happen, assuming that the system works?"

Answers to these questions will also provide you with a sense of what is most important to the users so that you can prioritize the testing.

Your responsibility in testing is to identify the defects you have discover, state why you consider them to be defects, and note the importance of fixing

them. Management makes the decision whether to fix the discovered defect based on your report. In addition to determining whether to fix it, management also decides when to fix it. When a defect impairs the solution, you have to show management that is the case. You cannot expect management to recognize the impact of a single defect.

User Acceptance Testing?

I was recently on a job in San Juan, Puerto Rico. The lead business analyst for the project told me that the policy stipulated that the users should test the systems for acceptance. On this particular project the tester happened to be the user who had specified the requirements for the system during requirements definition. When he tested the resulting product, he noted a number of functions that were missing and actions that did not perform the way he wanted. None of the cited defects were in the requirements that were used to create the system. There were things he had missed during requirements definition and others he discovered while he was testing, and some new ideas about the system that occurred to him while he tested. However, all items were reported out of testing as defects. Management demanded that the project team explain why there were so many defects; the project team blamed the business analyst for not "getting the requirements right the first time"; political unrest ensued; and so it goes.

In other situations, after thorough testing and sign off by the users, defects were found in production. When the defect surfaced, the testers acknowledged that they had not thought of those tests, because, after all, they are not professional testers. And that is true. It is not their job. When the business assigns a couple of users to leave their day job and spend a few days testing the system, the users test whatever they want to test based on their own personal experience. Often the users who do the testing were not present when the requirements were specified and did not contribute to the problem or solution definition. They are also not given much more instruction than, "Check out the system and see if it will work for us." Rarely are they given a copy of the requirements document, and even more rarely will the requirements actually represent what is being delivered, not having been updated since sign-off.

We want the users' input and review of the software and process they are going to have to live with, perhaps for the rest of their business lifetime. We cannot abandon our role as product champion and let the blame fall on the user for failing to test appropriately. As a business analyst, you must stay fully engaged during acceptance testing even when the users are doing the testing.

On the other hand, the business analyst must resist the temptation to exclude the users from testing altogether and perform the testing on the users' behalf, showing them the results for approval when the tests are over. Since you, as business analyst, are not a regular user of the system day-in and day-out, you are most likely going to miss a number of tests that the users might think of. And it is those tests that you want to have executed during acceptance testing.

🔍 Example

Keep Your Testing to Yourself

Fran, a business analyst for a large insurance company, describes a situation where her company bought a commercial off-the-shelf product for sales. The business analysts participated in the selection of the vendor and were assigned to do the acceptance testing. The systems people were in the room while the business analysts were testing on behalf of the users. The business analysts did not think it was intuitive and friendly to the users. As they discovered and pointed out problems, the IT people were saying, "No, that's easy" or "That's not a problem." While believing that they represented the users fully and accurately, the business analysts were cautious and finally called in the actual users from the home office, who tested it again and agreed with the business analysts that the system was not appropriate. The system was ultimately rejected. The IT group groused that the business analysts had prejudiced the users to reject the package, which further exacerbated the gap between IT and business. At one point the business analysts were told by the business analyst manager, "You are being negative because you do not want this system."

The rejection of the vendor system was somewhat traumatic for all, especially the upper-level manager who authorized the purchase. The systems representatives may have felt their obligation was to support management's decision, or they may have been exhibiting classic IT arrogance about the users. The business analysts may have been predisposed against the vendor's offering and unduly influenced the users into rejection before the users actually tested. Regardless of the possible reasons, the reactions of the system people who observed the testing increased the time of the overall process (the second round of testing with the users) and exacerbated contention between IT and the business community, which is not alleviated easily.

TABLE 16.1 Responses to Identified Defects

Requirements	Software	Test Cases	Action
Correct	Correct	Correct	None
Correct	Correct	Defective	Change the test case
Correct	Defective	Correct	Fix the code
Correct	Defective	Defective	Fix the code and change the test case
Defective	Correct	Correct	Fix requirements
Defective	Correct	Defective	Fix requirements and test case
Defective	Defective	Correct	Fix requirements and code
Defective	Defective	Defective	Fix all

Handling Defects

During acceptance testing the business analyst receives and/or reports the discovered defects, by comparing the discovered defect to the requirements or solution document. This is because the defect might exist in:

- The requirements (requirement is written wrong, or is incorrect).
- The software (software does not match requirements or has an operational defect).
- Test case (test case does not expose an existing defect or erroneously identifies an action as a defect).

Table 16.1 shows how one organization determines the disposition of defects found during testing. They assume that the defect is not always in the software, but could exist in the requirements, or in the test cases themselves.

Whether users are involved or not during the acceptance-test stage, make sure you can demonstrate that the acceptance criteria, as defined by the problem owner or other stakeholders, have been achieved: Prove the problem has been solved. When you can do that, it becomes difficult for the business community to demand anything additional before accepting the solution. The business community may want additional features or functionality, and such requests are considered—for the next project or release.

Testing Does Not Stop at Delivery

The job is over for the testers and QA when the product has been approved for delivery. They have proven that the product is ready. They have

identified deficiencies that have either been corrected or that management has decided to accept, at least for the time being.

The business analyst continues to test even after the product is in production in the business community. You are not running additional acceptance test scenarios at this point. You are observing operations and verifying that the new changes are properly installed and used. Mostly though, you are measuring the results, either manually or through evaluation devices embedded in the system or process. You are confirming that the solution is returning the expected value to the organization.

Our responsibility as business analysts is to make sure the problem is completely solved and that it stays solved. We evaluate the results in production to identify additional problems. As Gerry Weinberg says, "If you have a solution that does not produce at least three new problems, you have the wrong solution."[1]

So now you have a system that is confirmed, tested, approved, and accepted. Job done, right? Not really. The solution still has to be used by the business community to solve the problem and there may be resistance, apathy, or outright sabotage from the user community if the change is not brought into production in the right way. The solution team brings about the change and the business analyst brings about the transition: transitioning the business community into the new processes and procedures. The next chapter discusses how to do that.

Note

1. Donald Gauss and Gerald Weinberg, *Are Your Lights On?* (New York: Dorsett House, 1990).

Transition and Change Management

It should be borne in mind that there is nothing more difficult to handle, more doubtful of success, and more dangerous to carry through than initiating changes.

—Nicolo Machiavelli

Too many IT projects successfully complete on time and within budget but deliver a product or solution that is misused by the business community, or simply not used at all. The solution has to work in the business environment. And it has to be used. The process workers have to be prepared to receive and incorporate the change into their daily rituals so that the problem is solved and stays solved. The primary focus of the business analyst during the final stages of solution development is to prepare the business for transition to the new process. This means making sure new features or functions are incorporated into the process and the business community or customers are ready to receive the changes. Then, after the solution has been approved for delivery, the business analyst works with the business to overcome resistance, engender a smooth roll out of the solution, and record the reactions, positive and negative, to the product in use.

Change is acceptable and condoned when the change can be seen to be appropriate, and managed by communication with those who are impacted or affected by the change. When the change appears arbitrary or capricious to those affected by the change, there is resistance, or the change is completely ignored. There may even be sabotage.

A project can be on time, within budget, and deliver everything that was asked for, and still not solve the problem. In fact, the product may not even

be used by the business in the end. The interface may be unwieldy, for example, a user interface designed by software developers[1] or the users like their old way of doing things better or the product may be a solution that is not really needed. These projects are called *failed successes* by University of Virginia professor Ryan Nelson, Director of the Master of Science in Management of IT program at the University of Virginia's McIntire School of Commerce.[2]

On the other hand, the project may be a disaster from a PMI perspective. It may be late, over budget, and not deliver everything that was promised. However, the features and functions that are delivered are absorbed immediately into the workplace and bring instant and measurable benefit to the organization. Professor Nelson calls these projects *successful failures*. One might consider that the difference between a successful project and successful change is another difference between the project manager and the business analyst (see Chapter 5 for discussion of all the differences).

The project manager is responsible for successfully producing a product that will change the organization; the business analyst is responsible for the successful transition of the organization to use the product and affect the change.

Taking it one step further, the solution team thinks in terms of applications, use cases, programs, classes, user stories, and objects. Success to the solution team is measured in efficient code, elegant design, and inventive use of technology. The business, however, does not care about code, systems, or even requirements. The stakeholders think in terms of competitive advantage, higher revenues, lower costs, and business problems solved. The business analyst has to think in terms of solutions to business problems, and the pieces as tools that facilitate the solution—the software, hardware, and data. In the end, the product that the project produces loses its individual identity and becomes just another function, another tool, that the businessperson uses to do their job. Once the product is in operation, the effort to create the product is forgotten. It is taken for granted, just like we forget that the buildings in which we work were once a mass of beams in a hole temporarily populated by construction workers.

The value of the tool that we define and then put into production increases with the number of workers who use the tool and with the volume of information handled by that tool. That is not to say that we choose our problems to solve by the size of the user population. A single vice president using a single application to evaluate the sales trends of the past five years may be able to increase sales by 30 percent, increasing the value of the organization significantly. On the whole, the more pervasive the solution is in the organization, the higher the value to the organization.

Regardless of the change being made, there will be affected parties. Even when the change seems like a simple change affecting a small

percentage of customers, such as changing the sales tax calculations on the corporate Web site for several states, there is still going to be impact. For example:

- The accounting people will have different numbers coming in.
- The state retail tax agencies will need assurances that the tax calculations are working properly for them.
- The legal or compliance departments will want assurances that the process meets all regulatory dictates.
- The customers from those states will see the prices of repeat purchases suddenly increase.

In each of these four cases, the acceptance tests prove that the new software works and the regression tests prove there are no side effects in the rest of the Web site sales processing. This is still not going to be enough. The project sign-off, when all the requirements are met, will not suffice for the constituencies mentioned.

Steps to Ensure Successful Change in the Organization

As business analyst, you are responsible for transitioning the changed process into the business community. John Kotter, one of the gurus of organizational change, defined eight steps for guiding successful change in an organization.[3] The business analyst is responsible for the success of each step.

1. Establish a sense of urgency for the change to take place.
2. Create a guiding coalition.
3. Develop a vision of the changed environment.
4. Communicate the change vision.
5. Empower action to affect the change.
6. Generate short-term wins for both business and project team.
7. Consolidate the gains and produce more change.
8. Anchor the new approaches in the culture.

Sense of Urgency

You are responsible for creating a sense of urgency with the problem statement. Stating the problem in negative or constraining language increases the sense of urgency. Maintain this sense of urgency throughout the solution life cycle by keeping the solution team and product stakeholders focused on the problem. Use this sense of urgency to initially get approval to solve the problem.

Coalition Guided by the Business Analyst

You are responsible for creating and maintaining the *coalition* of the business—the product stakeholders, IT, the solution team, and management, the problem owner, executive decision maker, and others. This coalition collaborates to create the successful solution and to ensure the successful integration into the business environment.

Creating and Communicating the Vision

You are responsible to get a definition of the vision (see Chapter 8). Work with the problem owner to define a scenario that describes the business environment when the problem is solved. Investigate the problem domain and discover the conditions that cause the problem, and define what is necessary to do to change those conditions. Refine the vision as you complete your investigation. Continuously communicate the changed vision to the coalition to get their feedback and keep them apprised of the progress of the solution.

Empower Action

You are responsible for empowering action to solve the problem. Use the solution document. The solution document, when approved, grants authority to the development team to begin constructing the solution. The document simultaneously empowers action to be taken, and instructs the solution team clearly and unambiguously on what must be done to solve the problem.

Establish Short-Term Wins

You are responsible for dividing the vision and/or solution into functional business goals that can be delivered incrementally to establish the early wins and provide feedback. These wins prove that the solution is on track. The feedback tells you and the solution team what else must be changed and identifies changes in direction. When the solution is such that it cannot be subdivided, you continuously communicate, display, show, or demonstrate the solution, as it is being developed, to gain the feedback and establish the short-term wins. These short-term wins may be:

- Smaller portions of the entire solution delivered incrementally.
- The results of a timeboxed delivery in an agile development approach.

- Successive releases of a product, with each release incrementally improving the baseline product with new features and functions and reduced defects.
- Prototypes the solution team demonstrates to the product stakeholders.

Consolidate the Gains

You are responsible for making sure that all parties are aware that the solution is in progress. Make sure that changes emanating from the feedback are incorporated into the product and into the solution document. When applicable, explain to product stakeholders why changes cannot be made. Work with the solution team to ensure the changes are added. Throughout the sometimes painful process of inclusions and exclusions that result from a partially delivered or reviewed product, keep the coalition intact and moving forward.

Anchor the Change

You are responsible for anchoring changes in the business environment. This may include a number of activities that you must handle directly or indirectly. Your responsibility is to make sure the business processes are changed appropriately to accommodate the system changes. Remember that the solution team is only responsible for the technical solution. You have to make sure the technical solution fits into the overall business process to achieve the business goals and objectives.

Anchoring new approaches is specifically a business analyst's job. IT will leave the project and go to another one once the solution has been accepted. Business analysts must stay with the product to make sure it is working as expected and continues to solve the problem. They will also record new problems that arise from the use of the new software changes: the bugs or defects as well as new enhancements that solve new problems.

Orchestrate the Transition

Your primary function in the role of change agent is to orchestrate the transition from the old to the new. Transition is the psychological preparation in the business for the change to the organization. There is no role in IT that performs the tasks necessary to prepare the business community. Further, since the business community is entrenched in the current situation and awaiting the change, they are not necessarily capable of transitioning themselves. In the book, *Managing at the Speed of Change*, Daryl Conner suggests that we need to reframe the thinking of the affected business

community and address not only what the changes are, but also how they can be achieved successfully.

The users want assurance that the problem has been solved in a way they can use. The development community is generally not equipped to provide this assurance because they do not know what the users' fears and hesitations are. Their attitude is, "We solved your problem as you requested, now use it." I don't mean this in a nasty way. It is simply that the development team assumes the users want the change and therefore should be accepting it with open arms as soon as development has it ready. However, delivering a new software system that changes the way people work in a business is not the same as completing a new addition on the house. Change is what IT does. Transition is the business analyst's job.

Many times the resistance you get when instituting a new order of things is valid. Change can be frightening and with good reason. As Alfred North Whitehead said, "The major advances in civilization are processes which all but wreck the society in which they occur." The changes to your organization may cause havoc with the established order of things. Those who embraced the business process that was replaced are finding themselves devoid of their expertise and perhaps expendable. The routine that the business was used to is now quite different. For many it will be like changing jobs. For some it will mean actually changing jobs. Is it any wonder that regardless of the benefits of the new system, there will be resistance, even from those who might have been considered champions of the change?

Facilitate the Transition

The primary concern you will face when reviewing the solution in production is how the system is being used. Check out how the users are interacting with changed user interfaces: Are they scowling and staring at the screen for minutes at a time? Do they revert back to the old way of doing things? Are they complaining? Are they suggesting new or alternate ways of doing things? Also check out how the solution fits in with the rest of the business process in the problem domain and look for other impacted business processes. Is the product being used to solve the problem as it was intended?

There are several ways of observing the solution in production without wandering around the hallways peering into cubicles.

In some organizations the business analyst takes a position on the help desk for a period of time immediately after the implementation. All calls about the new system or process are sent to the business analyst. In this way the users get the most informed person about the system, and the business analyst gets to hear firsthand the problems and issues users are having with

the system. Of course, getting positive feedback is not likely. Not many users take the time to call the help desk to report how great the new system is.

In many organizations, the business analyst trains the users in the new system or trains the trainers to train the users. In other organizations, the business analysts write the user's documentation. As one business analyst told me, "The business area doesn't want to write application documentation, and when technical people do it no one can understand it." In both cases, the business analyst must get closer to the users to be able to successfully complete the task.

Some companies are adopting a warranty period. This period is the first thirty to ninety days after release into production. During this time resources are retained, specifically the business analyst, in case anything goes wrong. The business analyst in these companies performs a number of tasks related to ensuring successful adoption and recording the reactions and issues. The project budget includes funds to cover the warranty period.

In the same vein, many companies are adopting a project extension. This is an additional period of time added on to the project schedule and budget to accommodate a transition period after the project has completed the implementation. Typically it is the business analyst and project manager who stick around full time for the extra week or two allocated. During this time, the business analyst specifically seeks feedback on the solution and elicits any improvements or other issues that may come up in operation.

When the solution is installed, the process workers should feel that they are in control of their job or at least have a semblance of control. The next sections provide more details on how to accomplish this.

What do you have to do to ease the transition?

- Train where necessary, with formal classroom training and/or on the job.
- Provide encouragement through hands-on experience.
- Reinforce the change and the benefits of the change.
- Empathize with the business community in their natural resistance to change and do what is necessary to gently overcome that resistance.
- Make adjustments and record the issues.
- Review and log the suggested improvements or modifications identified by those who are using the changes.
- Celebrate the success.

Training

One main advantage of the business analyst doing the training, as is done in some organizations, is that the business analyst gets a lot of feedback from the users in a controlled setting. This feedback usually provides the

framework for subsequent releases. Another consideration is that a training session may be the only way to get all the users together in one place at the same time.

Regardless of whether you do the training or not, you are responsible for the process workers readiness to use the new process and/or product when it is put into production.

In a Virginia-based consulting company, the business analysts take on the task of training the users when the new releases are ready for production. Since the users are spread out geographically, representatives from each region are sent to the training and then return to their bases and train the other users. The training session serves also as a beta test, in which the users express their concerns with the new features or their appreciation. From this training session, the business analysts typically get many new feature suggestions and requests for future releases.

Manuals

The business analyst is also responsible for making sure that there is sufficient documentation available to the users of the system. This might mean two-inch-thick tomes that sit on the back of the desk, or it might mean on-line help. Either way, the materials that are provided to the users and process workers must be checked, verified, and tested. The measure of the value of such materials is the number of help desk calls that could have been answered with the standard developer response, "Read the manual."

Help Desk

Someone has to make sure the help desk is prepared to receive questions and complaints about the new feature just installed. It is embarrassing for everyone when the help desk first hears about a new feature from a user calling with a question after the feature is in production. Make sure the help desk has everything it needs to support the users on Day 1.

Determine what the most likely questions the user community will have during the first several hours they are operating the new system or process. Provide the answers for those questions to the help desk so they are prepared to respond to the onslaught.

Timing the Change

Determining when to effect a change is an important criterion for successful change. The business analyst helps upper-level management determine when the time is right to make a change. Basically the organization should

change when it can no longer afford the status quo; when the cost of maintaining the status quo is greater than the price of transition.

When is the best time to make the change and put the new process into production? Are there any restrictions on when the new process can be deployed? In a retail company, December might not be the best time to roll out a new point of sale system. However, many manufacturing companies close their operations during the end-of-year holidays; this is an ideal time to bring up new functionality.

Major and Minor Changes

Change is disruptive on many levels. It is not to be taken lightly by the business analyst. The business analyst often acts as the voice of reason amidst the chaos of change. The more information and interaction the business analyst can promulgate, the easier it is for the change to be assimilated.

There is a tendency for the business analyst and upper-level management to categorize change as major or minor. It may not be done officially. Based on the budget for the change or the number of process workers involved, the change is allocated to a certain position of importance in the scheme of things. The PMO and other governance groups generally categorize projects this way when they are determining which business case to accept.

Regardless of the level of change determined by those determining such things, keep in mind that what is important is the workers' perspective. What you might consider a minor change to a data entry screen for a small set of process workers may well be considered a significant major change to those workers. It may be the only change they have seen in a number of years, so by comparison it is big to them. You have been changing processes and systems all around them, so this seems relatively minor to you. When you act nonchalantly about the change, the process workers may perceive that you are not listening to their concerns about what they perceive as a major disruption in their work lives.

The larger the change and the more impact it has on the organization, the longer the transition period needs to be. People have to be aware the change is coming and be reminded of it on a regular and consistent basis so that they can start incorporating the change into their routine before it happens.

Do Not Change a Thing

Not only must the business analyst understand what has to be changed to solve the problem, the business analyst must also know what must be left

intact. This requires impact analysis to determine the overall impacts to the organization of any change.

Typically, tbusiness management expects a laser-like change to be made to solve the problem; a change that has no other impact than to fix the situation, a change that requires little or no alteration in operations. The solution team, however, unless otherwise directed, will not feel so constrained. Considering the typical aggressive deadline placed upon them, the solution team will produce the simplest solution that will work, disregarding operational impacts to other business areas. Many times it is indeed easier and a more cost efficient approach to retrain the process workers than to create a less intrusive solution.

Figure 17.1 diagrams the impact. The business expects the change to be a small impact at the center of the problem domain. The actual impact will generally be larger as shown by the lighter shaded area in the background. Typically there will be additional impact, expected or unexpected, that may cut across the problem domain and into neighboring constituencies circle.

The business analyst's role is to define what needs to be changed in the current business process *and* what must be left intact. The business analyst provides guidance to the solution team for the change scope and constraints, based on the ability of the organization to incorporate the change.

The new system (or significantly modified old system) affects other systems and business processes around it, those that feed information to the newly changed process, and those receiving information from the new process. This causes them, in turn, to make changes to accommodate the new system. That, in turn, may cause the new system to have to change accordingly. Because a vicious cycle can develop with change–impact–
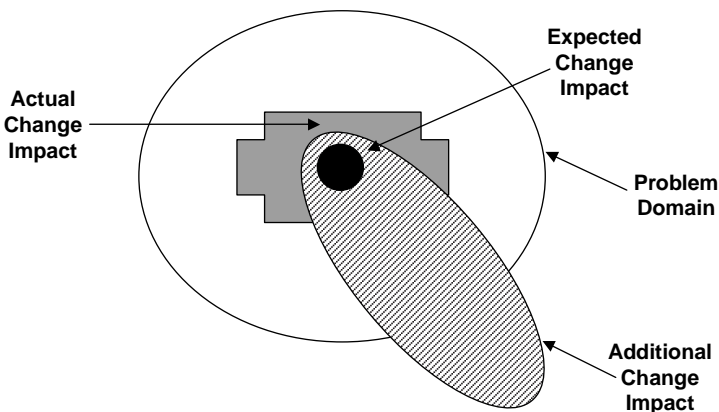
FIGURE 17.1 Impacts of Change

change–impact, identify and accommodate as many impacts on surrounding systems and business processes as possible at the start. But remember that changes will occur.

The change agent may lean toward a strategy of bringing in the change over a period of time instead of in one big bang. We've all been through organizational changes where upper-level management announces that things will be different on a certain date, and it all happens, or perhaps not. A better approach is to implement parts of the change and gain feedback from those affected to improve the next implemented part in an incremental fashion.

Wrapping Up

The solution is up and running and the process workers are using it successfully. The problem is solved. The celebratory party is over and the hangovers have subsided. There are still a few more tasks a business analyst has to do to wrap it up and move on. This is when the concept of business functional goals as discussed in Chapter 9 comes in handy.

Updating the Baseline Requirements

As shown in Figure 17.2, once the solution has been produced and accepted into the business community, the final requirements that constitute the description of that solution are incorporated into the baseline requirements for that product. Ultimately, the baseline requirements should be a fairly exact description of the actual system in operation. This will include use cases, screen shots, and other diagrams that augment the written requirements and are meant for persistent documentation of the requirements. A requirements management tool, such as Requisite Pro, DOORS, or Calibre RM, may be used as a repository for the baseline.

Why do we want to do this? Simple—to make the work easier the next time anything in this problem domain changes and you are called in to define the change. When that happens, you do not have to create a new problem domain description and model as was described in Chapter 12. It is already done. You can use the current baseline to identify the conditions that are causing the new problem.

Think ahead to the next change you have to make to this product, and if you are already on the lookout for that change, you know that you will be measuring the problem domain to establish the productivity baseline. To make it easier, incorporate the measurements into the supporting computer system or into the business process.
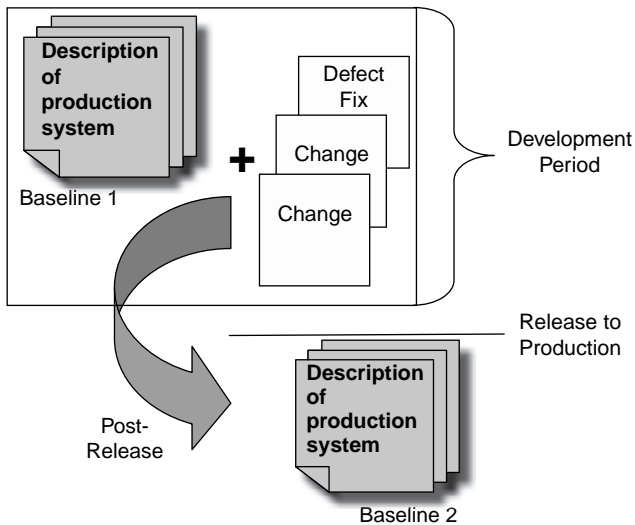
FIGURE 17.2 Update Requirements Baseline

The Evolution

Remember that a problem is the difference between the way things are and someone's perception of the way things should be. The way things are now is with the solution installed and in use in the business community.

The project is a point in time, a unique exercise. The product is a continuing, evolving process, changing with the changing business environment, improving with use. The users of a system will find better ways of using it through experience, ways not conceived of by you or anyone on the solution team. And you, the business analyst, are the guiding light for the evolution of the product over time, until events overtake the product, as technological and business environments change to the degree that the product is retired and replaced by a newer model. And most likely, you will be there to define what the business wants done with the new technology.

What you are looking to hear are comments like, "How hard would it be to . . . ?," "It might be easier if you . . . ," "This doesn't feel right . . . ," and so forth. Each represents a problem expressed by a process worker working with the solution. Each may also represent the next problem-solving effort and a new project. And management may ignore every one of the comments. Your job is to capture the comments and bring the applicable ones to management as new problems.

Each new problem is a new challenge, a new opportunity for the business analyst to add value to the organization.

Notes

1. Alan Cooper devotes an entire book to this issue, called *The Inmates Are Running the Asylum* (Indianapolis: Pearson Education, 2004).

2. Karen Klein, ''Lessons in the Rearview Mirror,'' ProjectsAtWork, www.projectsatwork.com/content/articles/233429.cfm (October 12, 2006).

3. John Kotter, *Leading Change* (Cambridge, MA: Harvard Business School Press, 1996).

Postscript: Where to Go from Here

Progress might have been all right once, but it's gone on too long.
—Ogden Nash

Future of Business Analysis

"There is no career path for business analysts. When business analysts want to make a move in their careers, they have to change jobs."

One View

Guy Beauchamp, in a white paper entitled "The Fundamentals of Business Analysis," makes these predictions:

- Projects will stop doing analysis. Lean and agile (and the now old RAD (Rapid Application Development) and JAD (Joint Application Development)) could be seen as the starting points for this. Their method to mitigate the risks of missing links in the chain of reasoning is to do lots of small releases so that the cost of correction is lower for each release.
- The Business Analyst profession will disappear into an in-fight of methodologies and rival accreditation schemes, lose all credibility and will be replaced by another profession (ironically having to do the same job under a different name as there is no rational way around the fact that in order to develop the right solutions the right analysis has to be done— eventually!). Of course, that profession will face the same issues as

Business Analysts did and . . . well, as George Bernard Shaw put it, "We learn from history that we learn nothing from history."

■ Systems Analysts will take over the Business Analyst role and develop analysis products fit for their purposes (developing computerized systems) and not necessarily fit for business purposes (developing solutions may be including computerized systems, business procedures, organizational units to operate procedures and so on).[1]

A More Positive View

I see a future where business analysts are promoted to executive-level positions because of their experience in the organization and their applicable knowledge.

As an example, a company in Sacramento has a centralized business analyst team led by a business analyst manager. They started with 20 business analysts who report to the business analyst manager, and not to either the business or IT. The company is run by a president (who is a doctor) and a board of doctors. They convene periodic meetings to review strategic direction, status of the company, and even the larger projects that are running, especially in IT. The meetings include the CIO, vice presidents, directors, and the senior legal staff. Seven months after the business analyst team was formally launched, the business analyst manager was invited to join the meetings to provide advice and counsel to the board from his perspective of what the organization was doing. Clearly, the board feels that the business analyst manager has insight into the organization that they can use in their decision-making about the future of the organization.

Why We Need Business Analysts

While there are considerable political and organizational challenges to the position and role of the business analyst, the importance of the business analyst to the organization is inestimable. The business analyst orchestrates successful change in the organization, creates a coordinated, collaborative team of business and technological personnel working toward a single goal, solves the organization's most important problems, increases revenue, and lowers cost. And the business analyst does this without wearing a superhero mask and flowing cape.

> *No company can make a business decision of any merit without kicking off some sort of related IT process.*
>
> —Michael Vizard

There is perhaps no single position in the organization more central to the organization's overall success outside the top executives—the CEO, CIO, CFO, and so on—than the business analyst. The business analyst identifies problems, separates the real problems from the symptoms, defines solutions to the problems, and brings technical and business staff together to collaboratively solve problems. The business analyst may be the only position in the enterprise that has knowledge of business processes as well as the IT technology that supports them.

The True Value of the Business Analyst

"Why on earth do we need business analysts?" [A question asked by a CFO during a presentation on the work several business analysts had done on a successful mission-critical project.]

Here is just a sampling of the ways in which the business analyst brings significant value to their organization and the people in it.

Ombudsman

Who do you call? In organizations that have full-fledged business analyst professionals operating in the center between problem and solution, the business analyst becomes the first call the business makes when faced with any issue beyond simple operational questions.

Business managers tend to call business analysts for advice and counsel when considering an initiative that may involve computer technology, or any technology. They generally find it easier to talk to the business analysts than the technologists. In their role as filter, the business analysts help the business manager determine whether the initiative is practical and feasible and perhaps offer alternative ways of solving the problem.

Want to know the value of the business analyst? Compute the time and money saved when the business analyst solves a problem or answers a question so that an IT project does not have to be executed. Compute the time saved when the business analyst can provide a quick answer or do the research to get the answer instead of the businessperson taking time away from production to do it. The result of this calculation gives you an idea of the value of a business analyst.

Helping Hand

At many organizations, the business analyst unit includes the help desk. This is not an unusual situation. Those on the help desk are learning all about the

problems that process workers have with the various systems in the organization on a constant basis. Who better to work with the business analysts than those who deal with the process workers' problems all day?

In some organizations, the business analysts regularly cycle through the help desk. Each business analyst spends some time on the help desk, usually just after the solution they are working on is delivered. This is part of the post-delivery routine. Some business analysts man the help desk during elicitation as a way of learning firsthand what the problems are with a given system or process. There is always one business analyst on the help desk.

In the Center of It All

The business analyst unit, reporting to IT, the business, or independently enfranchised, provides a central repository of business process information, business system documentation, enterprise-wide knowledge of business practices, and process improvement and information system change efforts. The business analysts, in their position between IT and the business units, are the center of communications between all groups.

In the U.S. subsidiary of a Paris-based global financial organization, the business analysts, working in a BA Center of Excellence, act as a clearinghouse to review all change proposals for any IT-related modification or enhancement. Similarly, the business analysts at a large New York–based U.S. bank also appraise new project proposals on behalf of the bank's very active PMO.

Want to know the value of the business analyst? Think about the number of times someone in IT or the business searched the organization for information about a particular system or business process in order to gather enough information to get the job done. Consider the times a seemingly successful IT project in one area of the organization caused several other areas increased work or problems, or the implementation generated a flurry of new change requests and trouble reports. Remove those issues from the organization and you have an idea of the value of the business analyst.

Increasing the Value of the Organization

In this day and age, it is hard to see how any organization can survive and prosper without the role of the business analyst. Like a good editor who catches the typo that would have changed the essential meaning of the main thesis of your paper, or the medical internist who identifies apparently unrelated symptoms to diagnose a disease that can be cured to save your life, the business analyst observes patterns in business processes, anomalies in process flow, warning signals that indicate problems are on the horizon, and identifies the cures or solutions that keep the business alive and help it prosper.

Where You Go from Here

To senior management and the executive suite, the business analyst has experience with executive-level decision making. The business analyst provides the information to upper-level management and can see firsthand how that information is turned into a decision. The business analyst has direct knowledge of the business processes that keep the revenue flowing and the organization healthy. The business analyst also has an intimate understanding of the computer systems and technology that support those business processes. The business analyst has spent time honing communication skills: negotiating, mediating, influencing, playing the diplomat, and establishing relationships over all levels of the organization. What better training can anyone receive in managing the organization in preparation for the executive suite?

Sounds fantastic? In an open letter to CIOs from the CIO Executive Council for *CIO* magazine, the Council recommended the following steps to lead their organizations. I have **bolded** each of the steps that include business analyst roles or responsibilities that have been presented in this book.

- **Emphasize your role as an "agent of change"** at the executive table.
- **Create agile processes and architecture** so your business can turn on a dime.
- **Look for solutions, ideas and partnerships beyond the traditional boundaries of your organization**.
- Tap the emerging resources of the global business community, social networks, and consumerized technology.
- **Originate commercial product and service ideas by studying market trends and external customer needs**. Too many of us undervalue these areas.
- Develop the next generation: Bring your senior staff along on this journey.[2]

The steps that the CIO Council recommends that CIOs take are those that are central to the successful business analyst's role: look for solutions, agent of change, create agile processes, and so forth. Based on this open letter, it appears that business analysts are CIOs-in-training. Performing the roles of the business analyst may put you in the next generation referred to in the last step.

While it may be a bit of wishful thinking, in 5 or 10 years I see business analysts joining the ranks of CIOs and even CEOs. I think this eventuality will be beneficial for the profession of the business analyst, the image of those in the executive suite, the economy, and the world in general.

The BA as Prognostician

Information technology achieves its value only through its application to business objectives.

—Marvin Richardson

The business analyst is more than just a diagnostician. You must see the problems before they occur. This calls for prognostication based on analysis. Just as an experienced physician can examine the results of tests and advise the patient that the onset of a disease is imminent unless precautions are taken, the business analyst can also tell when aspects of the business are not healthy.

IT produces a lot of facts and data, not necessarily intelligence. Management is too busy making decisions to gather and assemble all the data from all the sources now available through technology.

You are the one who has hands on the data, familiarity with what is available, and knowledge of the decisions that have to be made. You can advise upper-level management of the information available to them and organize it to present high-level views and summaries that support the quick and informed decisions needed in today's fast-moving business world.

The business analyst is not simply a researcher responding to requests of upper-level management to provide information. The business analyst often initiates the actions that identify problems in the organization. This is the promise of the business analyst: understanding the organization and its processes and identifying where the processes are not as effective or efficient as they should be.

Because the individual business analyst is limited in scope by his or her own personal experience within the organization, a business analyst group, formal or informal, provides a continuing cross-reference of insight into the organization and its processes. This enables each business analyst to share the experiences of all business analysts and the commonalities appear more often.

Power to the Business Analyst

Power is not a possession, but a relationship.
    —*Gerald Weinberg,* Becoming a Technical Manager

As a business analyst you do have a lot of explicit power. You are vested with the power that comes with creating solutions for problems. You have the power of information and knowledge. You have the power that accrues to the one who is in the center of business change. You have the power deriving from the vast number of relationships you develop as you successfully perform your duties.

And anyone who can do these things for the organization possesses great power: the power of being the go-to person for solving problems.

The business analyst might likely be the most powerful role in the organization. While in the scheme of things in today's organizational structure, that power will rarely be explicit, it is clearly inherent in what you do on a day-to-day basis. Your job is to directly create outcomes of value to the organization. Every solution you define increases the organization's value and enhances your value to the organization.

Laura Brandenburg has a Web site for business analysts called Bridging the Gap (www.bridging-the-gap.com) and has two e-books on the subject. Here is her manifesto for business analysts:

Out of chaos, we create order.

Out of disagreement, we create alignment.

Out of ambiguity, we create clarity.

But most of all, we create positive change for the organizations we serve. [3]

Doug Engelbart, a computer pioneer, contributed the mouse, among other things, to the world. He described technology as an augmenter of human intellect. Let me paraphrase that description and apply it to the business analyst: The business analyst increases the capability of the organization to approach a complex problem situation, to gain comprehension to suit the organization's particular needs, and to derive solutions to problems.

That is our mission as business analysts. And we have certain rights that are essential to completing that mission.

As a business analyst you have the right to:

- Ask questions.
- Understand the problem and problem domain first.
- Make sure you are solving the right problem.
- Challenge the business.
- Challenge the solution team (to explain why they are choosing to solve the problem in a certain way).
- Come up with solutions.
- Define the problem domain.
- Ask more questions
- Solve the problem.

While it may be a relief to business analysts to realize that they have these rights as part of their profession, business analysts must also

acknowledge that along with the rights, they have the responsibility to do all these things as a professional business analyst.

Granting rights might not be enough, or might not resonate with you or your management. You might say that informing your manager or the business stakeholders that you have the right to define the problem domain or to ask questions might only draw a blank stare and a comment reminding you of your place in the scheme of things—at least their concept of your place. So, let's talk about empowerment. You may not feel you have the power in the organization or even in the department to really complete the mission or manifesto. You certainly are empowered to do more than to record requirements that are gathered from the users, and certainly more than to act as subject matter experts on behalf of the businesspeople who are too busy to respond on their own. And the business analyst does more than take the blame when the business demands too much, or when the solution team does not satisfy all the requirements.

As a business analyst, you are empowered to:

- Ask questions.
- Challenge the norms—the way things are done—both on the business side and on the development side.
- Try new techniques, methods, and processes to perform your job.
- Suggest new methods or ways of executing business processes in the organization.
- Analyze instead of accept.
- Get and understand information first before committing to a solution.
- Ask more questions.
- Do the good quality job for the organization as a whole instead of only individual organizational entities.
- Define and solve the business problem.

So, when your mother-in-law, or Aunt Susan, or the nosy neighbor down the street, or the mildly interested fellow at the cocktail party asks what you do, and you answer, "I'm a business analyst," and they stare blankly at you awaiting an explanation that does not involve use-case diagrams, elicitation, ROI analysis, and other explanations that cause glazing of eyes and shifting of feet backwards away from you, simply tell them you:

- Identify business problems that need solving.
- Solve business problems.
- Ensure the solutions bring value to the organization.

And that is the essence of the business analyst.

Notes

1. Guy Beauchamp, "Business Analysis—Delivering the Right Solution for the Right Problem," www.smart-ba.com/articles/ba_chain_of_reasoning.pdf (accessed 10/29/08).

2. CIO Executive Council, "Why It's Time for Us to Refocus," *CIO* 44 (January 1, 2009).

3. www.bridging-the-gap.com/about-bridging-the-gap/business-analyst-manifesto/.

Business Analyst Process

This appendix provides an overview of the solution development process that a business analyst follows, from identification of a problem in the business community to the implementation of a solution in that same business community. Each of the activities is discussed at length in the text of the book.

I. Define the Problem and Product Scope

1. Define the problem owner.
   - The person or department that has the authority to seek a solution to the problem; can identify or describe the real problem; can approve the problem definition and can voice or approve the vision.
2. Prepare an information-gathering plan to determine the problem.
   - What information is needed to define the real problem?
   - Where can the information be found?
   - How can the information be acquired?
   - What is the order of acquisition of the information?
3. Elicit information about the problem.
   - Talk to the problem owner and managers and observe the problem domain to determine what the real problem is.
4. Analyze the information.
   - Make sure the stated problem is really a problem.
   - Make sure the problem is aligned with the organization mission and strategies.
5. Determine the real problem to be solved.
   - Make sure the problem is the real one that needs to be solved and that it is the correct problem.
   - Make sure the solution of the problem satisfies all issues presented by the customer (manage expectations).

6. Confirm with the problem owner (or executive decision maker).
   - Make sure that the real problem that has been defined is the one that management wants solved.
   - When management needs justification to solve the problem (or when organizational policy dictates), perform the requisite analyses: ROI analysis, cost/benefit analysis, and/or feasibility studies.
7. Define the product scope.
   - Obtain the vision of the solution and acceptance criteria from the problem owner.
   - Determine the product stakeholders.
   - Identify the business risk of not solving the problem and the potential impact to the organization when the problem is solved and any other identified business risks.
   - Determine the justification for solving the problem.
   - Identify any business or product constraints that may be imposed on the solution.
   - Identify any functional goals or business objectives that must be achieved for solving this problem. Break the product into incremental deliveries based on functional goals where possible.
8. Create a formal or informal decision document.
   - Combine the product scope with other required information and produce a business case and/or project charter or other document(s) that the organization may require to make a final decision to solve this problem.

II. Define the Solution

1. Prepare an information-gathering plan to determine the solution.
   - What information is needed to define the solution?
   - Where can that information be found?
   - How can the information be acquired?
   - What is the order of acquisition of the information?
   - Classify user communities when they are large.
   - Identify hidden, indirect, and disadvantaged users when possible.
2. Elicit information about the problem domain.
   - Understand completely why the problem exists and what conditions cause the problem.
   - Use information-gathering techniques, such as interviews, meetings, use case sessions, observation, and so on.
   - Confirm that you completely and accurately understand the information the process workers provide.
3. Analyze the information to determine potential solutions for the problem.
   - Categorize and filter the information and requirements.

- Model or diagram the problem domain, business processes in the problem domain, and the applicable environment.
- Identify the conditions in the problem domain that are causing the problem.
- Analyze and model the solution.
  - Data-intensive systems might be modeled with an entity relationship diagram.
  - You might model process-intensive systems with a data flow or activity diagram.
  - Systems with high user interaction might be best rendered in use cases.
- Continually confirm your analysis with the affected product stakeholders that the part of the solution that affects them will work and is acceptable.

4. Document the solution.
   - Record the results of your analysis so you can verify them with the product stakeholders wherever they are.
   - Confirm with the stakeholders that the solution completely and accurately solves the problem.
     - Get parts of the solution confirmed as they are defined.
     - Check technical and project feasibility with the solution team.

5. Write the solution document.
   - Render the solution in a form that is understandable to the product stakeholders and business management, and is in a form acceptable to the solution team.

6. Validate the requirements with peers.
   - Use peer review or inspection as the format to validate.
   - Review the requirements document with the solution team.

7. Get the solution document approved by the executive decision maker.

III. Keep Requirements Up-to-Date throughout Software Development

1. Review systems requirements and/or design to identify changes to requirements or variations from the defined solution.
   - When the solution varies from the definition and business expectations without technological rationale, suggest conforming to the business expectation of the solution.
   - When the solution justifiably varies from the definition and business expectations, change the solution document accordingly and confirm the change with the affected product stakeholders.

2. Create a process to review any changes to the evolving system that may affect requirements.

IV. Prepare Acceptance Tests Based on the Defined Acceptance Criteria That Will Prove to Business Analyst and Stakeholders That the Problem Is Solved

1. Write acceptance test scenarios, scripts, or cases that prove the problem is completely solved.
   - Tests should be executable by users or users' representatives.
   - Results should be understandable by users or product stakeholders.
   - Final results should be understandable by problem owner or business management to prove problem was solved.
2. Participate in the acceptance-testing phase.
   - Execute the tests and share the results with the stakeholders.
   - Supervise the users or users' representatives in the execution of the acceptance testing.
   - Work with quality assurance or quality control to ensure that tests prove the problem is solved.
   - Record changes, alterations, and modifications to the requirements as a result of software changes due to acceptance testing (there should be very few).
   - Record suggested or recommended changes for post-release consideration.

V. Enable the Transition of Solution into Production

1. Prepare the business community for the solution.
   - Make sure process workers get appropriate training and documentation where necessary.
   - Remove, reduce, or identify any final resistance to the solution in the business community so the solution can be given a fair chance to succeed.
2. Evaluate the solution in production.
   - Observe the solution in use to ensure that the problem is solved completely.
   - Record any suggestions for improvement or defects that are reported during the first period of use.

VI. Start Over Again

The Principles

The newest computer can merely compound, at speed, the oldest problem in the relations between human beings, and in the end the communicator will be confronted with the old problem, of what to say and how to say it.

—Edward R. Murrow

Essence: As in most occupations and roles in the organization, there are a number of principles that apply to the successful completion of the job. Some of the principles are high level and generic while others are specific to an area of concern. The following is a list of such principles that apply to the role of business analyst.

The business analyst is a role that can be played by any job or position, from developer to upper-level management, and must be a role that is played in any successful business problem-solving effort. The tenets in this book are aimed at those who must play that role so they may play the role successfully. The following represents a summation of those tenets and principles.

Principle 1: Focus on the Product

The Business Analyst Focuses on the Product, Not on the Project

The project is the domain of the project manager and the solution team. The product is the result of that project. The business analyst starts with the end in mind[1] and keeps it focused. The business analyst's responsibility is to solve the problem, and get that product into the business environment.

Understand What a Solution Is Worth to the Business

At any time the business analyst can tell anyone the value of the change being made to the organization, and for the most part to any component of that change. Each feature and function created by the solution team plays a part in the overall solution and each has its intrinsic value. The business analyst always knows why.

Principle 2: First Define the Problem and Then the Solution

Requirements Describe the Solution to a Defined, Understood, and Approved Business Problem

Here is a complaint from a senior business analyst: "Most projects are in design-mode long before they have established what the problem is that they are trying to solve has been defined. Too often I see project teams discussing how the screen is going to look and what push buttons are going to do before anyone knows what business problem we are trying to fix."

The set of requirements the business analyst creates describes the solution to the business problem in terms the business community can understand and that they agree will completely and accurately solve their problem. The technical community can also understand what needs to be done so they can completely and accurately create the software to solve the problem.

Principle 3: Users Do Not Have Requirements

Users Do Not Have Requirements; Stakeholders Do Not Have Requirements—They Just Have Information

The business analyst seeks information and analyzes it to produce the solution document containing the requirements.

Principle 4: Focus on Information Not Individuals

Start by determining what information you need to solve the problem. Use the information-gathering plan to help structure the information-gathering process. Always keep the focus of the elicitation on gathering information.

No part of the solution should be based on only one source; all parts of the solution should be verified, confirmed, and validated, preferably by someone or something other than the source of the information that produced that part of the solution.

Principle 5: Separate Elicitation from Analysis

When Eliciting Information, Do Not Analyze; When in Analysis, Do Not Create Information

While eliciting information, the business analyst focuses only on getting as much information as possible. Analyzing the information as it is acquired appears as though the business analyst is judging the responder and will stem the flow of information. During analysis, the opposite is true. Any new information created while analyzing is a business analyst's assumption. The facts only come from the business.

Principle 6: Improve the Process First then Add Technology

Evaluate Non-IT Solutions First before Resorting to Computers and Software to Solve the Business Problem

Since most business analysts come from IT, there is a natural tendency to assume that all business solutions can be solved with information technology and that the only solution to a business problem is by applying the use of computers. Many times, however, there are much more elegant and simple solutions to business problems: changing processes, relocating process workers, redistributing the work, and so forth.

Focus on the Business, and How IT Can Be Used to Improve and Enhance the Business's Status Quo

Look for human solutions rather than technical ones. IT will come up with the technical solution to support the human business interaction. The business analyst has to make sure that human beings can comfortably use the technology. Keeping focused on the human aspects of the solution keeps the business analyst focused on the business as much as the technology and balances the solution.

Constantly Review and Appraise the Organization's Processes and Operations to Determine Where Changes Can Be Made That Will Add Value to the Organization

Every solution is the source of new problems. Every problem has ancillary and dependent problems. Focus first on the real problem to solve now, and keep your eyes open for other problems that exist or that may exist when a solution is applied.

Take a holistic view of the organization and apply inductive reasoning to the environment surrounding the stated problem to discover any other problems. By looking at the whole problem domain instead of only focusing on the immediate issue, the business analyst:

- Gets a wider view of the problem in context.
- Identifies ancillary problems and issues.
- Gets a better view of the impacts that may attend a given solution.
- Is able to grasp different views of the problem and the conditions that cause the problem.

Principle 7: Communicate, Cooperate, Collaborate

Keep Communications Flowing in All Directions

Step out of the way of the communication and let it flow naturally. That is, let the solution team talk to the product stakeholders and vice versa. Only step in between to clarify, ameliorate, document (as in taking notes), facilitate (as in moving the discussion along), or mediate.

Do what is necessary to promote the flow of information among all parties of the solution. Do not, however, force reticent team members to communicate against their personality or coerce information from recalcitrant individuals.

Live on the Feedback

Organize your communication efforts around obtaining feedback from all parties. Keep announcements, status reports, and one-way communiqués to a minimum. When one of your communications does not receive feedback, consider that your communication failed and seek another way of transferring the information more successfully.

Checkpoints

There are a few checkpoint meetings that you want to hold formally or informally with one or more participants. The primary purpose of the checkpoints is to make sure the solution is moving in the right direction.

The checkpoints are not status meetings or this-is-what-I-have-done meetings.''

Problem Owner Confirmation

Perform a quick check of the defined problem statement with the problem owner to make sure the problem is correct and it is the problem that should be solved. At that time you ask three questions:

1. Is this the problem you want solved?
2. What is your vision of the solution?
3. What do you need to see to know we have solved the problem?

Checkpoint Alpha—Confirmation of Product Scope

This checkpoint is a preview of coming attractions for the solution team, assuming there is one, and/or the product stakeholders. The purpose of the meeting is to validate the starting point for the solution life cycle and get an initial verification of product scope feasibility. The questions to be answered at this checkpoint are:

- Is this product scope feasible?
- Is there anything I missed?
- Does this make sense?

Checkpoint Beta—Confirmation of Good Requirements

The second meeting, which could be informal or formal, comes after the good requirements have been confirmed, when you have the solution to the problem that the process workers and/or customer agree to, and before you start the validation process of turning the solution into a formal set of requirements or solution document. The purpose is to technically confirm the solution and uncover any technical infeasibility, and to let the solution team know what is coming.

The questions you ask at this checkpoint are:

- Is there anything in the solution I missed?
- Does this solution make sense?
- Is it still feasible?
- Do you understand what the solution document is saying?

Checkpoint Charley—Review of Design

The third meeting, which is formal, is done after the technical team has defined a technical solution. The technical team presents its solution to the

business analyst. The business analyst makes sure the solution still solves the business problem and updates the business solution to include any variations that result from the technical solution. The questions to be answered at this checkpoint are:

- How does the technical solution solve the business problem?
- Are there any technical changes that affect the solution document?
- What do I need to change in the solution document to keep the document synchronized with the solution?

Do Not Let Documentation Substitute for Communication and Collaboration

The requirements documentation should be the distillation or results of the process and all the communication that is part of the process. Confirmation activities should not be focused on getting the document approved, but on getting the solution correct. Approval will then follow naturally.

Principle 8: The Business Analyst Owns the Solution Requirements

Once you have defined the solution and the business community has agreed that it solves their problem, you are the only one who has the authorization to physically change the approved requirements. When you change them, the person(s) who signed the original document should also agree with, if not re-sign, with the changes made.

Requirements Are Written for Those Who Create the Solution, Not for Those Who Have the Problem

The final target audience for the requirements is the solution team. While the users or stakeholders need to agree that the requirements completely and accurately solve the problem, the requirements are written for those who are actually going to build the solution.

The Solution Document Must Match the Delivered Solution

Throughout the development process, there will be changes to the product. There will be trade-offs during systems analysis and design. There will be technical changes due to the build process. Testing may introduce changes.

While the business analyst must evaluate all the changes in light of a valid solution to the problem, many changes will be acceptable or unavoidable. The business analyst's obligation is to modify the requirements to reflect the changed vision of the product.

The requirements state the solution to the business problem. The system delivered must do what the requirements say. The system delivered must solve the problem.

There is certainly a lot of discussion in the agile communities about the value of a formal set of requirements. There is no argument among the agilists about the value of a well-maintained and documented code base. Business analysts derive the same benefit from documenting and maintaining complete requirements, as developers do from maintaining their code base.

Analyze, Analyze, Analyze

Your last name is analyst and everything you do is about analysis, from preparing decision papers for upper-level management to analyzing problem statements to determine what the real problem is. You do not accept anything as fact without analyzing to make sure it is fact. Your ability to analyze is what separates you from the requirements recorders.

Principle 9: Gain Acceptance as Well as Approval

Getting the solution document approved by the appropriate authorities on the business and solution sides is not enough for the business analyst. All must accept the solution document. That means the product stakeholders accept the changes that are going to be made to their environment, and the solution team understands and accepts the statement of solution and agrees that they can affect the solution.

Principle 10: Make the Business Community Ready for the Product

You do not want to create a solution that nobody uses. The solution might be elegant and satisfy all requirements; however, when it is not used, it fails as a solution. The business analyst makes sure the product stakeholders—those affected by the problem and those impacted by the solution—have the requisite training and documentation, and are prepared for the change to their environment.

Principle 11: Measure Twice, Cut Once

Measure the problem domain to establish the depth and breadth of the problem as part of our justification. Then implement the change. Once the problem is solved, perform the exact same measurement so that you can show the improvement.

Place the post-implementation measurement into the product. In this way you can continue to measure to ensure the product continues to solve the problem and provide an indicator of future problems. It also establishes your first measurement the next time you change the system.

Measuring also provides a continuing justification and proof that the proposed changes are worthwhile and increases the trust that management has in you as a business analyst.

Note

1. Stephen Covey, *The Seven Habits of Highly Effective People* (New York: Free Press (Simon & Schuster), 1989).

Why We Do Not Get Good Requirements

The following list of comments, complaints, and concerns was collected over many years from hundreds of business analysts all over the world. It does not seem to matter how long the business analyst has been practicing or where in the world he or she works, or what kind of organization they work in, the comments tend to be the same. This list is certainly not all of the obstacles to getting good requirements. Knowing that the obstacles you face are similar in nature to those faced by most other business analysts may be comforting or it may be frustrating. Either way, it is better if you also have some means or techniques for overcoming those obstacles. I provide a key that connects to these techniques in the book.

TABLE C.1 Why We Do Not Get Good Requirements

Comments	Answers Found
Not enough time.	Chapter 11
Don't listen to the business owners.	Chapters 7, 11
Don't know the problem.	Chapter 12
Inability to communicate.	Chapter 11
Assumptions.	Chapter 11
Unrealistic expectations.	Chapter 7
Users and management give us solutions, not requirements.	Chapter 11
Users communicate symptoms.	Chapters 8, 11

TABLE C.1 (*Continued*)

Comments	Answers Found
Users don't tell us everything.	Chapter 11
Users/management don't focus on requirements.	Chapter 7
Not knowing if you have captured everything.	Chapter 9
Don't know what a good requirement is.	Chapters 11, 13
User's cynical attitude in light of past system failures.	Chapter 11
Users don't realize what they do—leads to omissions and lack of detail.	Chapter 12
Management doesn't know what's really going on.	Chapters 12, 7
Don't know if users will use it right.	Chapter 12
Deal with the least valuable person.	Chapter 11
Jump to conclusions.	Chapter 13
Changing business environment.	Chapters 8, 17
Lack of consensus on the customer side.	Chapters 7, 11
Requirements too general.	Chapter 14
Not specifying business rules.	Chapter 14
Resistance from users.	Chapter 11
Combining tech and non-tech vocabulary.	Chapter 14
Conflicting requirements from customer/user.	Chapter 11
Miscommunication.	Chapters 5, 11
Non-technical users.	Chapter 7
Not defining terms up front.	Chapter 5
Requirements person has too much knowledge.	Chapter 11
Users think it is a waste of time.	Chapter 11
Egos on the part of users or systems analysts.	Chapters 7, 11
Conflicting user goals and requirements.	Chapters 11, 13
Unspoken requirements.	Chapters 13, 14
Users change requirements after approval.	Chapter 17
Too focused on how, not enough on what.	Chapter 13
Information silos.	Chapter 13
Users don't read the documents—they just sign off.	Chapter 14
Needs change over time.	Chapter 17
Lack of commitment/support from sponsor.	Chapter 7
Poor user's skills to use system.	Chapter 7
Cultural differences.	Chapter 11
Lack of implementation training.	Chapter 7
Co-lateral duties on part of business.	Chapters 7, 11
Stakeholders forced to think in engineering terms.	Chapters 7, 11
Users don't share information with management and other users.	Chapter 11

TABLE C.1 (*Continued*)

Comments	Answers Found
Use system to change business processes/operations.	Chapter 10
Requirements are a moving target.	Chapters 12, 14
Getting requirements from just one person.	Chapters 7, 11
Client doesn't understand what's currently in use.	Chapter 12
IT thinks they know the business.	Chapters 6, 15
Clients lose interest before the requirements are done.	Chapters 11
Lack of commitment from business.	Chapters 7, 11
Users are too busy.	Chapter 11
Uncertainty of going to a new world.	Chapters 17
Not understanding of impacts on other systems.	Chapters 12, 13
System is overly complex—no one really understands it.	Chapter 12
Not getting agreement.	Chapter 13
Stakeholders not validating.	Chapter 13
Not getting a decision.	Chapter 13
User availability.	Chapter 11
Not getting to the right people.	Chapter 11
Not getting users until after system is installed.	Chapter 11
Prevented by management from getting the requirements because decision was made.	Chapter 11
Manager gets the wrong requirements from the user and gives them to us.	Chapter 11
Users are not competent in their jobs.	Chapter 7
They change the problem on us.	Chapter 8
Users fear replacement.	Chapter 11
User doesn't know what they want.	Chapter 11
Resistance.	Chapter 11
Lack of understanding of environment.	Chapter 12
Lack of management support.	Chapters 7, 11
Politics.	Chapter 10
Government mandates changes.	Chapters 10, 13
Users don't know what they want.	Chapter 11
The BAs don't know the business.	Chapter 12
Lack of a standard process.	Parts 3, 4, 5
Scope creep.	Chapter 9
IT speaks a different language.	Chapter 5
Businesspeople don't know what to tell us.	Chapter 7
Misunderstandings and misinterpretations.	Chapter 11

TABLE C.1 (*Continued*)

Comments	Answers Found
Suspicions on the part of users about why we are asking questions.	Chapter 11
Users not giving us realistic requirements.	Chapter 13
No one knows the big picture.	Chapter 12
Changing users/customers.	Chapter 11
Hidden agendas.	Chapter 11
No input from the user.	Chapter 11
Defining the how, not the what.	Chapter 13
Improper levels of expertise.	Chapter 11
Communication inconsistency.	Chapter 5
User does not have all the information up front.	Chapters 11
Lack of client knowledge.	Chapters 11
Having an open-ended scope.	Chapter 9
Need to get into coding right away to show some progress.	Chapter 15
Inexperienced requirements person.	Parts I, II
Communications between developers and users are not open.	Chapter 11
Documentation (requirements) not maintained throughout life cycle.	Chapter 17
Developers don't read requirements before they start developing.	Chapter 15
What is balance between over-documentation and not enough?	Chapter 14
Over optimism by everyone but developers.	Chapter 11
Not enough questions.	Chapter 11
Lack of technically sophisticated users.	Chapters 7
User knows what they want, but can't communicate it.	Chapters 7, 11
Developers give users products they don't want.	Chapter 15
Communication problems.	Chapter 5
Poor buy-in.	Chapters 8, 9, 11, 14
Different expectations.	Chapter 7
Changing priorities.	Chapters 13
Quality assurance not involved soon enough.	Chapter 16
People are confused about requirements.	Chapters 11, 15
Users want everything, just in case.	Chapter 11
Business doesn't want to do requirements.	Chapter 11
Client doesn't understand his or her own process.	Chapter 12

TABLE C.1 (*Continued*)

Comments	Answers Found
Client already has solution.	Chapter 11
Continually changing while development is going on.	Chapter 15
Lack of cooperation due to fear or other reason.	Chapter 11
Requirements influenced by personal objectives.	Chapter 11
No consequences of failure; nothing happens if we don't solve the problem.	Chapter 8
Lack of upper-level management support.	Chapters 9, 10
Proprietary issues: security, privacy, territoriality.	Chapter 11
Lack of confidence in BA by users.	Chapter 11
Users resistant to change.	Chapters 7, 11
Users work against you.	Chapter 11
A process already exists that satisfies the requirements.	Chapter 12
Lack of knowledge—no SME or SME does not know.	Chapters 7, 11
Problem not completely defined.	Chapter 8
Scope not defined.	Chapters 9
The users don't tell us everything.	Chapter 11
The SME isn't an expert.	Chapter 7
We don't get good requirements because we do not know the problem.	Chapter 8
We do not get good requirements because in the time it takes to get the project approved, the problem changes.	Chapter 8

Comparison of the Roles of Business Analyst, Systems Analyst, and Project Manager

"Since I am doing all three roles, what is the difference between the project manager, the systems analyst, and the business analyst?"

As shown in Table D.1, all three roles (the business analyst, project manager, and system analyst or technical lead) basically perform the same activities with few exceptions. This is where all the confusion arises. A business analyst does risk analysis and so does the project manager and system analyst. The project manager defines scope, but so do the business analyst and system analyst.

The difference among the three roles is one of focus. The business analyst focuses his activities and tasks on the business or product. The project manager focuses her activities and tasks on the project. The system analyst focuses his activities and tasks on the technical aspects of the solution.

The project manager has the overall authority and responsibility for the project and its success. The system analyst or technical lead is responsible for the technical aspects of the implemented solution. The business analyst is responsible to ensure that the originally defined business problem has been solved completely and at the expected quality.

TABLE D.1 Comparison of Roles

	Business Analyst	Project Manager	Systems Analyst
Primary communication	Communicate with **business** and **product** stakeholders and the project manager.	Communicate with everyone.	Communicate with the **technical** and solution teams.
Define solution	Define solution to **business** problem.	Define solutions to **project** problems.	Define solutions to **technical** problems.
Identify problem	Identify **business** problem.	Identify **project** problem(s).	Identify **technical** problem(s).
Provide justification	Justify the problem-solving effort by defining or creating the **business** plan, C/BA, ROI analysis, **project** charter.	Justify the **project** and changes to the **project** scope: time or resources.	Justify the solution design and **technical** changes.
Scope definition	Define **Product** Scope.	Define **Project** Scope.	Define **Technical** Scope.
Address trade-offs	Confirm **business** trade-offs or trade-offs that affect the **product**.	Handle **project** trade-offs, e.g., budget, schedule, scope, risk.	Identify design or **technical** trade-offs.
Create change	Be an agent of change for the organization.	Be an agent of change for both **business** and **technical** aspects.	Be an agent of change for **technical** aspects and architecture.
Analyze and/or manage risk	Define and manage **product** risks and risks to the **business**.	Define and manage risks to the **project**.	Define and manage **technical** risks.
Plan	Plan the requirements definition.	Plan the entire **project**.	Plan the **technical** solution.
Monitor and control		Monitor and control staff and resources.	

Analyze and manage stakeholders	Define and manage **product** stakeholders.	Define and manage **project** stakeholders.	
Manage expectations	Manage customer, user, and **business** expectations.	Manage management (IT and **business**) expectations.	Manage **technical** expectations.
Identify requirements	Identify **business** requirements (What to do).	Identify **project** requirements: resources, skills, etc.	Identify system and **technical** requirements (How to do it).
Analyze impact	Analyze **product** and **business** impact.	Analyze **project** impact.	Analyze **technical** impact.
Test	Test the **product** (Acceptance testing).	Test the **project** plan.	**Technical** testing.
Evaluate alternatives	Evaluate **business** solution alternatives.	Evaluate **project** alternatives.	Evaluate **technical** alternatives.
Estimate	Estimate the time and effort to define the **product**.	Estimate the time and effort to successfully complete the **project**.	Estimate the time and effort to develop the **technical** aspects of the implementation.
Perform project close activities	Conduct a **business** analyst retrospective and a requirements definition lessons learned.	Conduct **project** close and retrospective **project** lessons learned.	Conduct **technical** retrospective and lessons learned.
Perform pre-deployment activities	Conduct **product** training and familiarization to ensure a smooth transition.	Manage **project** turnover activities.	Define **technical** changes necessary for deployment (migration, conversion, etc.).
Perform deployment activities	Assess organizational readiness to receive **product**.	Manage **project** cut-over into **product**ion.	Handle **technical** issues during cut-over.

(continued)

TABLE D.1 (*Continued*)

	Business Analyst	Project Manager	Systems Analyst
Perform post-deployment activities	Assess use of **product** in the **business** environment.		
Mediate	Mediate **product** disputes among **business**, IT, and upper-level management.	Mediate **project** disputes among team members and others.	Mediate **technical** issues among team members or other **technical** personnel.
Obtain sign-off	Obtain sign-off for solution document and for final deliverable **product**.	Obtain sign-off for **project** completion.	Obtain sign-off for **technical** solution.
Analyze	Analyze **business** processes and status.	Analyze **project** processes and status.	Analyze **technical** processes and status.
Report	Report status of **product** and solution to **business** and **project** manager.	Report status of **project** to upper-level management and others.	Report status of **technical** solution to **project** manager and others.

APPENDIX E

Context-Free Problem Definition Questions

1. What is the problem?
2. What is the business justification for solving the problem?
3. What are the risks associated with the issues?
4. What if we don't solve the problem, or don't solve it within the deadline, if a deadline is stated?
5. What are the impacts to the business for any given solution?
6. Are there any business constraints?
7. Who is affected by the problem?
8. Who owns the problem?
9. When does the problem occur (intermittent, constant, chronic, etc.)? How long has it been going on?
10. What does it look or feel like when the problem is solved (what is the vision)? How will we know that the problem is solved? Where in the organization does the problem exist?
11. How (when) do you know it's a problem?
12. What is the alignment of the problem? What business strategy, objective, and so on is the problem or opportunity related to?
13. Who is the executive decision maker or sponsor of this project?

List of Nonfunctional Requirements Categories

The following is a list of typical nonfunctional or supplementary requirements (most are defined in IEEE Standard 1233, 1998):

- Reliability (e.g., mean time between failures, MTBF).
- Availability (e.g., expected hours of operation).
- Maintainability (e.g., ease with which components can be replaced).
- Performance (e.g., must return prompt within two seconds).
- Accessibility (e.g., different navigation paths for novice and experienced end users).
- Environmental conditions (e.g., dirty, dark, or dusty environments).
- Ergonomic (e.g., use of specific colors to reduce eye strain).
- Safety (e.g., loudness of signals, so they can be heard but not harm hearing).
- Security (e.g., who is authorized to do what).
- Facility requirements (e.g., require special electrical or phone capabilities or use standard already in building).
- Transportability (e.g., weight limits of handheld units).
- Training (e.g., are tutorials or textbooks required?).
- Documentation (e.g., online help, reference manuals).
- External interfaces (e.g., support industry-standard protocols).
- Testing (e.g., support remote diagnostics).
- Quality provisions (e.g., minimum required calibration intervals).
- Policy and regulatory (e.g., government requirements, constraints).
- Compatibility to existing systems (e.g., must support analog phone lines for Internet access).

- Standards and technical policies (e.g., must conform to SAE, ASME, or national electrical codes).
- Conversion (e.g., will support data from older versions of system).
- Growth capacity (e.g., will support X end users over Y years).
- Installation (e.g., can the new system be put into service while old is still running?).
- Migration (moving data to different platforms).

Here are some additional nonfunctional requirements categories that also apply to current problem-solving efforts:

- Accountability (compliance with federally mandated financial reporting and accounting regulations).
- Auditability (ability to produce information for an outside agency).
- Traceability (traceable to source of information, usually for audit reasons).
- Globalization (ability to be used in all languages, cultures).
- Localization (ability to switch from a generic view of a process or information to a view localized to a geographic or demographic constituency).
- Privacy (limitations on the distribution and/or access of information).
- Accessibility also now refers to provisions for the physically disadvantaged in accessing computer facilities, especially Web sites [as described in the U.S. American Disabilities Act (ADA) section 508 and the World Wide Web Consortium (W3C)].

Bibliography

Ambler, Scott. *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process.* Hoboken, NJ: John Wiley & Sons, Inc., 2002.

Ambler, Scott. "One Size Fits None." *Agile Modeling Newsletter* (January 2005).

Ambler, Scott. "Rethinking the Role of Business Analysts: Towards Agile Business Analysts?" *Agile Modeling* (2006).

Ambler, Scott. "Detailed Requirements Specification: Possibly a Worst Practice?" *ISSIG Review X* 2 (2006): 17–18.

Ambler, Scott. "Scaling On Site Customer," *Dr. Dobbs Journal* (January 2008): 63–66.

Ambler, Scott. "Agile is Relative." *Dr. Dobbs Journal* (March 2008): 65–67.

Andrews, David, and Kenneth R. Johnson. *Revolutionizing IT: the Art of Using Information Technology Effectively.* Hoboken, NJ: John Wiley & Sons, Inc., 2002.

Astels, David, and Granville Miller and Miroslav Novak. *A Practical Guide to Extreme Programming.* Upper Saddle River, NJ: Prentice-Hall, 2002.

Baker, Bud. "Filling the Gaps." *PMI Network* 22(5) (June 2008): 26–27.

Ball, Tad. "How Can IT Help Serve the Underserved Business Analyst?" *CIO* (September 29, 2008). www.cio.com.

Barrett, Kathleen, and Raed Haddad. "The Analysis Business." *Projects at work* (March 9, 2006). www.projectsatwork.com.

Beauchamp, Guy. "The Benefits of Business Analysis." *Modern Analyst* (November 9, 2008). www.modernanalyst.com.

Beauchamp, Guy. "Business Analysis—Delivering the Right Solution for the Right Problem." *Smart BA* (February 20, 2007). www.smart-ba.com/articles/ba_chain_of_reasoning.pdf.

Beauchamp, Guy. "The Fundamentals of Business Analysis." *Business Analyst Solutions* (2007). www.businessanalystsolutions.com/.

Beck, Kent, and Cynthia Andres. *Extreme Programming Explained: Embrace Change.* Reading, MA: Addison-Wesley, 2004.

Beck, Kent. "Don't Justify—." *IEEE Software* 24(2) (April 2007): 63.

Bittner, Kurt, and Ian Spence. *Use Case Modeling.* Reading, MA: Addison-Wesley, 2003.

Blaine, et al. "Software Quality Requirements." *IEEE Software* 25(2) (March 2008): 24–26.

Blais, Steve. "The Hat Trick." *All PM* (April 9, 2007). www.allpm.com.

Blais, Steve. "Casing out the Problem." *All PM* (April 9, 2007). www.allpm.com.

Blais, Steve. "Is Project Management a Dead End Job?" *All PM* (August 15, 2008). www.allpm.com.

Boegh, Jorgen. "A New Standard for Quality Requirements." *IEEE Software* 25(2) (March/April 2008): 59–62.

Borland Corporation. "Best Practices for Driving Strategic Value as an Analyst." (white paper, October 2008).

Brennan, Kevin. *A Guide to the Business Analysis Body of Knowledge.* Version 2.0. Toronto, Ontario: International Institute of Business Analysis, 2009.

Brooks, Fred. *Mythical Man Month: Essays on Software Engineering.* Reading, MA: Addison-Wesley, 1975.

Brooks, Fred. "No Silver Bullet: Essence and Accidents of Software Engineering." *IEEE Computer Magazine* 20(4) (April 1987): 10–19.

Brown, Craig. "Bloom's Taxonomy and the Business Analyst." *Better Projects* (March 15, 2007). www.betterprojects.com.

Brown, Craig. "How To Be an Agile Business Analyst." *Modern Analyst* (September 7, 2008). www.modernanalyst.com.

Buckingham, Marcus, and Curt Coffman. *First Break All The Rules.* New York: Simon & Schuster, 1999.

Button, Brian. "Double Duty." *Better Software* (February 2006): 14–23.

Cantor, Murray. *Object-Oriented Project Management with UML.* New York: John Wiley & Sons, 1998.

Cherubin, Sam, and Kimberly Terribile. "Business Analysts: A Key to Companies' Success." *CIO* (August 12, 2008). www.cio.com/article/443340/business_analysts_a_key_to_companies_success.

Chhatpar, Arun. "The Evolving Role of the Business Analyst." *IBM Developerworks* (June 3, 2008). www-128.ibm.com/developerworks.

CIO Executive Council. "Why It's Time for Us To Refocus." *CIO* (January 1, 2009 C): 44. www.cio.com/article/471412/why_CIOs_Need_to_Focus_their_Attention_on_Growth.

Clements, Paul, and Linda Northrup. *Software Product Lines: Practices and Patterns.* Boston: Addison-Wesley, 2002.

Clements, Paul, and Linda Northrup. "A Framework for Software Product Line Practice, Version 5.0." Software Engineering Institute: Carnegie Mellon University, 2008. www.sei.cmu.edu/productlines/frame_report/req_eng.htm.

Cockburn, Alistair. *Agile Software Development*. Boston: Addison-Wesley, 2002.

Cohen, Steven. *Negotiating Skills for Managers*. New York: McGraw-Hill, 2002.

Conner, Daryl. *Managing at the Speed of Change*. New York: Random House, 2006.

Constantine, Larry, and Lucy Lockwood. *Software for Use*. New York: ACM Press, 1999.

Covey, Stephen. *The Seven Habits of Highly Effective People*. New York: Free Press (Simon & Schuster), 1989.

Creel, Christopher. "Requirements by Pattern." *Software Development* (December 1999).

Crispin, Lisa. "You Don't Say." *STQE* (September 2003). www.lisacrispin .com/wordpress/articles/.

Crosby, Phil. *Quality Is Free*. New York: Signet, 1980.

Culmsee, Paul. "The one best practice to rule them all—Part 1." *Clever Workarounds Blog* (2/12/09). www.cleverworkarounds.com.

DeMarco, Tom. *Structured Analysis and System Specification*. New York: Prentice Hall PTR, 1979.

DeMarco, Tom, and Tim Lister. *Peopleware: Productive People and Teams*. New York: Dorsett House, 1987.

De Jager, Peter. "Tempering Traumatic Transitions." (2006). www .technobility.com/docs/article094.htm.

De Jager, Peter. "Defending against Entropy." (2006). www.technobility .com/docs/article090.htm.

Dorner, Dietrich. *The Logic of Failure: Recognizing and Avoiding Error in Complex Situations*. New York: Basic Books, 1997.

Dyer, Sue. "Adverse Conditions." *Projects at Work* (March 9, 2006). www .projectsatwork.com.

Evans Data Corporation. "The New Business Analyst: A Strategic Role in the Enterprise." *Evans Data Corporation* (2006). www.compuware.com/ pressroom/news/2006/6068_ENG_HTML.htm (accessed 11/24/08).

Fairfield Solutions, Ltd. "The Benefits of Business Analysis." Fairfield Solutions (August 20, 2008). www.smart-ba.com/articles/benefits_of _business_analyst.pdf.

Feig, Nancy. "The Rise of the Business Analyst and Other Emerging IT Roles." *Bank Systems and Technology* (January 1, 2007). www .banktech.com.

Frost, David. "Interview." *USA Today*, January 23, 2007.

Gauss, Donald, and Gerald Weinberg. *Exploring Requirements: Quality Before Design*. New York: Dorsett House, 1989.

Gauss, Donald, and Gerald Weinberg. *Are Your Lights On?* New York: Dorsett House, 1990.

Geiger, Jonathan. "Intelligent Solution: Establishing a Center of Excellence."
    BI Review (March 20, 2007). www.bireview.com/article.cfm>articleid
    =222.
Gibbs, Mark. "Aiming for Perfection." *Network World* (April 21, 2003). www
    .networkworld.com/columnists/2003/0421backspin.html.
Gibbs, Mark. "Dealing with Hippos." *Network World* (September 24, 2007).
    www.networkworld.com/columnists/2007/092007-backspin.html.
Gilb, Tom. "Metrics Say Quality Better than Words." *IEEE Software* 25(2)
    (March 2008): 64–67.
Gilbert, Steve. "Wearing Two Hats: Analyst-Managers for Small Software
    Projects." *IT Pro* 6(4) (July/August 2004): 34–39.
Gladwell, Malcolm. *The Tipping Point: How Little Things Can Make a Big
    Difference*. Boston: Back Bay Books, 2002.
Goldsmith, Robin, and Dorothy Graham. "The Forgotten Phase." *Software
    Development* (July 2002). www.gopromanagement.com/PDF/07sd
    .Gold45–47.pdf.
Goldsmith, Marchall. *What Got You Here Won't Get You There*. New York:
    Hyperion, 2007.
Gottesdiener, Ellen. *The Software Requirements Memory Jogger*. Salem, NH:
    GOAL/QPC, 2005.
Goulston, Mark, MD. "The Care and Feeding of Your Career, How to
    Avoid Bumping Heads between IT and Business Managers." *CIO*
    (November 1, 2005). www.cio.com/article/13515/How_to_avoid_
    bumping_heads_between_IT_and_Business_managers.
Gruman, Galen. "Strategies for Dealing with IT Complexity." *CIO*
    (December 1, 2007): 52. www.cio.com/article/158356/strategies_for_
    dealing_with_IT_complexity.
Hall, Payson. "Changing the Hand You're Dealt." *Better Software*,
    (December 2006) http://www.stickyminds.com/sitewide.asp? Objec-
    tId=11710&Function=DETAILBROWSE&ObjectType=MAGAZINE
    &sqry=*Z%28SM%29*J%28MIXED%29*R%28relevance%29*K%28
    simplesite%29*F%28Payson+Hall%29*&sidx=42&sopp=10&sitewide
    .asp?sid=1&sqry=
    *Z%28SM%29*J%28MIXED%29*R%28relevance%29*K%28simplesite%
    29*F%28Payson+Hall%29*&sidx=42&sopp=10.
Hammer, Michael, and David Champy. *Re-Engineering the Corporation: A
    Manifesto for Business Revolution*. New York: Harper/Collins, 1993.
Hammer, et al. "Doing Requirements Right the First Time." *Crosstalk* 11(12)
    (December 1998).
Hass, Kathleen, and Richard Vander Horst. *From Analyst to Leader:
    Elevating the Role of the Business Analyst*. Leesburg, VA: Management
    Concepts, 2008.

Hass, Kathleen. *Getting It Right: Business Requirement Analysis Tools and Techniques*. Leesburg, VA: Management Concepts, 2008.

Hass, Kathleen. *The Business Analyst as Strategist: Translating Business Strategies into Valuable Solution*. Leesburg, VA: Management Concepts, 2008.

Hass, Kathleen. *Professionalizing Business Analysis: Breaking the Cycle of Challenged Projects*. Leesburg, VA: Management Concepts, 2008.

Hayes, Frank. "Survival Skills." *ComputerWorld* (September 5, 2005). www.computerworld.com/s/article/104355/Survival_Skills.

Head, Beverley. "Bridges over Troubled Waters." *CIO* (August 8, 2007). www.cio.com.au/index.php/id;889859495; pp; 1.

Highsmith, Jim. "Mission Possible." *Software Development* (July 1996). www.jimhighsmith.com.

Hohman, Luke. *Journey of the Software Professional*. Englewood Cliffs, NJ: Prentice-Hall, 1997.

Holmes, Neville. "In Praise of Professional Precision." *IEEE Computer* 41(1) (January 2008).

Holmes, Neville. "The History of the Computing Profession." *IEEE Computer* 42(2) (January 2009): 111–112.

Holmes, Neville. "The Craft of Programming." *IEEE Computer* 41(5) (May 2008): 90–92.

Holmes, Neville. "Agility and Respect." *IEEE Computer* 42(7) (July 2009): 98–100.

International Institute of Business Analysis. *A Guide to the Business Analysis Body of Knowledge* Rel 1.4, version 1, 2004. www.TheIIBA.org.

International Institute of Business Analysis. *A Guide to the Business Analysis Body of Knowledge*, (version 1.6), 2006. www.TheIIBA.org.

International Institute of Business Analysis. *A Guide to the Business Analysis Body of Knowledge* (version 2.0), March 31, 2009.

Kasse, Tim. *Practical Insight into CMMI*. Boston: Artech House, 2004.

Keen, Jack. "Don't Ignore the Intangibles." *CIO* (September 1, 2003).

Kirsner, Scott. "Lights, Camera, Innovation?" *CIO* (November 1, 2008): 28–30. www.cio.com/article/454917/what_CIOs_Can_Learn_From_Hollywood.

Klein, Gary. *Sources of Power: How People Make Decisions*. Cambridge, MA: MIT Press, 1998.

Klein, Karen. "Lessons in the Rear View Mirror." *Projects at Work* (October 12, 2006). www.projectsatwork.com/content/articles/233429.cfm.

Klein, Karen. "Enter the Business Analyst." *Projects at Work* (July 12, 2007). www.projectsatwork.com/content/articles/237229.cfm.

Koch, Christopher. "Go Ape." *Koch's IT Strategy Blog* (January 2007). advice.cio.com/itstrategy.

Kotter, John. *Leading Change*. Cambridge, MA: Harvard Business School Press, 1996.

Krutchen, Philippe. *The Rational Unified Process: An Introduction*. 3rd Edition. Upper Saddle River, NJ: Addison-Wesley, 2003.

Leffingwell, Dean, and Don Widrig. *Managing Software Requirements*. Boston: Addison-Wesley, 2003.

Lewis, Bob. "Advice Line." *InfoWorld* (September 16, 2002). www.infoworld.com/blogs/bob-lewis.

Liu, Jiming. "Computing as an Evolving Discipline: 10 Observations." *IEEE Computer* 47(5) (May 2007).

Longo, Terry. "Defining Roles Cycle in the BA Life." *Business Analyst Times* (September 2008). www.batimes.com.

Luft, Joseph. *Group Processes: An Introduction to Group Dynamics*. 3rd Edition. Mountain View, CA: Mayfield Publishing, 1984.

Lundquist, Eric. "The Skills that Really Matter." *eWeek* (December 1, 2008): 46.

Marasco, Joe. "Software Requirements." *Dr. Dobb's Journal* (September 2007): 10.

Martin, Chuck, Peg Dawson, and Richard Guare. *Smarts*. New York: Amacom Press, 2007.

Maxted, Suzanne. "The Importance of Business Domain Modeling." *Business Analyst Times*. www.batimes.com/articles/the-importance-of-business-domain-modelling.html.

McConnell, Steve. *Rapid Development: Taming Wild Software Schedules*. Redmond, WA: Microsoft Press, 1996.

McGregor, John. "Domain*." *Journal of Object Technology* 3(7) (July–August 2004): 71–81.

Ozkaya, Ipek, Bass, Len, Nord, Robert L. "Making Practical Use of Quality Attribute Information." *IEEE Software* 25(2) (March/April 2008): 25–30.

Paul, Debra, Donald Yeates, and James Cadle. *Business Analysis*, Swindon, UK: British Informatics Society, 2006.

Project Management Institute. *The Guide to the Project Management Body of Knowledge*. 3rd Edition. King of Prussia, PA: Project Management Institute, 2004.

Project Management Institute. *The Guide to the Project Management Body of Knowledge*, 4th Edition. King of Prussia, PA: Project Management Institute, 2008.

Podeswa, Howard. *UML for the IT Business Analyst*. Boston, MA: Thomson Course Technology PTR, 2005.

Pohl, K., G. Bockle, and F. van der Linden. *Software Product Line Engineering*. Berlin: Springer, 1998.

Preston, Ron. "IT Pros Still Undervalued." *Network Computing* (November 1, 2002).

Rackham, Neil. *SPIN Selling*. New York: McGraw-Hill, 1988.

Rising, Linda. "The Product Champion." *STQE* 5(3) (May/June 2005): 44–48.

Robinson, Pamela. "On the Case." *PMI Network* 20(10) (November 2006).

Rooney, Dave. "Embedded Collaboration." *Practical Agility Blog* (November 26, 2008). practicalagility.blogspot.com/2008_11_26.

Ross, Robert. "The Business Rules Approach." *IEEE Computer* 43(5) (May 2003).

Runeson, Per et al. "What Do We Know about Defect Detection Methods?" *IEEE Software* 24(4) (May 2007).

Sammy, Julian. "Different Roles for Business Analysts." *IIBA Executive Blog* (January 6, 2009). community.iiba.org/welcome.htm.

Schreiner, Keri. "The Bridge and Beyond: Business Analyst Extends its Role and Reach." *IP Pro* (November/December 2007): 50–54.

Schwaber, Carey and Rob Karel. "The New Business Analyst." *Forrester Research* (April 2008). www.forrester.com/rb/Research/new_business_analyst/q/id/43178/t/2.

Schwaber, Ken. "Defining Process vs. Problem Solving." *Application Development Trends* (March, 1996): 76–82.

Schwaber, Ken. *Agile Software Development with SCRUM*. New York: Prentice-Hall, 2001.

Sheard, Sarah. "Life Cycle of a Silver Bullet." *Crosstalk* 16(7) (July 2003).

Smith, Adam. *An Inquiry into the Nature and Cause of the Wealth of Nations*. CreateSpace, 2009, Seattle, WA.

Software Engineering Institute. *Capability Maturity Model Integrated/Software*, v1.1 (2002). www.sei.cmu.edu/index.cfm.

Software Engineering Institute. *Capability Maturity Model Integrated/Software*, v1.2 (2006). www.sei.cmu.edu/index.cfm.

Starr, Daniel. "Here's the Problem." *Projects at Work* (July 6, 2006). www.projectsatwork.com.

Stone, Douglas, Bruce Patton, Sheila Heen, and Roger Fisher. *Difficult Conversations: How to Discuss What Matters Most*. New York: Penguin Books USA, 2000.

Thayer, Richard, and Marlin Dorfman. *System and Software Requirements Engineering*. Los Alamos, CA: IEEE Computer Society Press, 1990.

Ury, Robert and Roger Fisher. *Getting to Yes*. New York: Penguin Press US, 1981.

Vizard, Michael. "Out of Sight, Out of Mind." *Baseline* (May 2007).

Vizard, Michael. "Take Control of Your Applications." *Baseline* (June 2007): 22.

Voas, Jeffrey. "Software's Secret Sauce: The '-ilities.'" *IEEE Software* 21(6) (November/December 2004): 14–15.

Wailgum, Thomas. "Why Business Analysts Are So Important for IT and CIOs." *CIO* (June 16, 2008). www.cio.com/article/336213/why_business_analysts_are_so_important_for_IT_and_CIOs.

Wailgum, Thomas. "8 Reasons Why CIOs Think Their Application Developers Are Clueless." *CIO* (September 3, 2008). www.cio.com/article/447189/8_reasons_why_CIOs_think_their_application_developers_are_clueless.

Wailgum, Thomas. "What Do Business Analysts Actually Do for Software Implementation Projects?" *CIO* (April 28, 2008). www.cio.com/article/343015/what_do_business_analysts_actually_do_for_Software_Implementation_projects.

Walsh, Katherine. "Hot Jobs: Business Analyst." *CIO* (August 15, 2007): 22. www.cio.com/article/120154/hot_jobs_business_analyst.

Weigers, Karl. "So What Are Requirements?" *Dr. Dobb's Journal* (September 2007): 20.

Weinberg, Gerald. *Becoming a Technical Leader.* New York: Dorsett House, 1986.

Weinberg, Gerald. "Just Say No: Improving the Requirements Process." *American Programmer* (August 1995): 19–23.

Whitten, Neil. "Nothing Personal." *PMI Network* 23(1) (January 2009): 28–29.

Wick, Angela, CBAP, PMP. Undated. "The Growth of a Business Analyst." www.bacollective.com.

Wiegers, Karl. *Software Requirements.* Redmond, WA: Microsoft Press, 1999.

Wurman, Richard S. *Information Anxiety 2.* Indianapolis, IN: Que, 2000.

About the Author

Steve Blais has been kicking around IT for over 43 years, from the days when it was called data processing, through the MIS era, and onward to the age of information technology. He has lived through the ups and downs of the computer technology industry, observing its cycles and recycles, fortunes and misfortunes, and tends nowadays to be quite bemused about the teeth gnashing, head slapping, and eyebrow furrowing associated with the relationship between technologists and organizational management.

Blais started in 1968 as a programmer for the U.S. Marine Corps and since then has done just about everything in the field, from programming to running software consulting companies. He has programmed, designed, and tested software; managed software development, managed projects, architected and administered networks; analyzed programs, processes, systems, networks, organizations, businesses, and even people.

He resides in Lakewood Ranch (Sarasota) and Key West, Florida. He provides consulting and training to companies all over the world in business analysis and management.

Index