Attached: The First Three Chapters, Plus More

Reviews

"Hiring the right people means the difference between success and faiure. It's not enough to hire 'good enough' . . . you need to hire the best, and nobody knows more than Johanna Rothman about that. This clear and comprehensive book joins *Peopleware* and *The Mythical Man Month* as must-reads for technical managers."

—Joel Spolsky
Founder, Fog Creek Software

"Rothman lays out the tasks and the issues, then addresses actual situations that might arise. She covers the entire subject thoroughly. . . .

"If you are a hiring manager in a high-tech field, you must read this book."

—Richard Mateosian
IEEE Micro

"If you are involved in any way with hiring techies, you need this book—not just as a one-time read, but as one you will refer to repeatedly."

—Earl A. Everett, Director of Engineering,
Vauban Advanced Technologies, posted on *Amazon.com*

"I'm not aware of any other book like this. It's a humane, yet tough-minded approach to hiring. Any technical manager who wants to hire well will be thankful for it."

—James Bach
CEO, Satisfice, Inc.

". . . practical, pragmatic advice on finding and hiring the right person. . . . full of examples, templates, and true stories that will help you make the best use of your time, fine-tune your hiring process, and hire the best."

—Esther Derby, President
Esther Derby Associates, Inc

About the Author

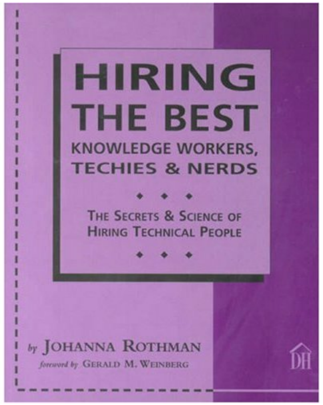
Johanna Rothman is a highly regarded speaker, author, and consultant; she is known for her pragmatic approach to the problems of managing high technology product development and workers. During the past twenty years, she has been influential in the hiring of hundreds of technical people, including developers, testers, technical editors, technical support staff, and their managers. Based in Arlington, Massachusetts, she is the president of Rothman Consulting Group (www.jrothman.com).

Partial Contents

Hiring the Best Knowledge Workers, Techies & Nerds

The Secrets & Science of Hiring Technical People

by Johanna Rothman
foreword by Gerald M. Weinberg



HIRING THE BEST
KNOWLEDGE WORKERS, TECHIES & NERDS
◆ ◆ ◆
THE SECRETS & SCIENCE OF HIRING TECHNICAL PEOPLE
◆ ◆ ◆
by JOHANNA ROTHMAN
foreword by GERALD M. WEINBERG
DH

ISBN: 0-932633-59-5
©2004 352 pages softcover
$43.95 (includes $6 UPS in US)

Proven Methods for Attracting, Interviewing, and Hiring Technical Workers

Good technical people are the foundation on which successful high technology organizations are built. Establishing a good process for hiring such workers is essential. Unfortunately, the generic methods so often used for hiring skill-based staff, who can apply standardized methods to almost any situation, are of little use to those charged with the task of hiring technical people.

Unlike skill-based workers, technical people typically do not have access to cookie-cutter solutions to their problems. They need to adapt to any situation that arises, using their knowledge in new and creative ways to solve the problem at hand. As a result, one developer, tester, or technical manager is not interchangeable with another. This makes hiring technical people one of the most critical and difficult processes a technical manager can undertake.

Hiring the Best takes the guesswork out of hiring and diminishes the risk of costly hiring mistakes. With the aid of step-by-step descriptions and detailed examples, you'll learn how to • write a concise, targeted job description • source candidates • develop ads for mixed media • review résumés quickly to determine Yes, No, or Maybe candidates • develop intelligent, nondiscriminatory, interview techniques • create fool-proof phone-screens • check references with a view to reading between the lines • extend an offer that will attract a win-win acceptance or tender a gentle-but-decisive rejection • and more.

You, your team, and your organization will live with the long-term consequences of your hiring decision. Investing time in developing a hiring strategy will shorten your decision time and the ramp-up time needed for each new hire.

Read more about this book at
www.dorsethouse.com/books/hire.html

Order Today! • (800) 342-6657 • (212) 620-4053 • fax (212) 727-1044 Phone, fax, or mail with credit card information, check, or money order. VISA MC AMEX
DORSET HOUSE PUBLISHING 353 WEST 12TH STREET NEW YORK, NEW YORK 10014 USA info@dorsethouse.com • www.dorsethouse.com

Library of Congress Cataloging-in-Publication Data

Rothman, Johanna.
Hiring the best knowledge workers, techies & nerds : the secrets & science of hiring technical people / by Johanna Rothman ; foreword by Gerald M. Weinberg.
p. cm.
Includes bibliographical references and index.
ISBN 0-932633-59-5
1. High technology industries--Employees--Selection and appointment. 2. High technology industries--Employees--Recruiting. 3. High technology industries--Personnel management. I. Title.
HF5549.5.S38R68 2004
658.3'11--dc22 2004016872

Quantity discounts are available from the publisher. Call (800) 342-6657 or (212) 620-4053 or e-mail info@dorsethouse.com. Contact same for examination copy requirements and permissions. To photocopy passages for academic use, obtain permission from the Copyright Clearance Center: (978) 750-8400 or www.copyright.com.

This publication is designed to provide accurate and authoritative information with regard to the subject matter covered. It is sold with the understanding that neither the publisher nor the author is engaged in rendering legal or other professional advice. If such assistance is required, the services of a qualified professional should be sought.

Cover Design: Nuno Andrade
Executive Editor: Wendy Eakin
Senior Editor: David W. McClintock
Editor: Nuno Andrade
Assistant Editor: Vincent Au

Distributed in the English language in Singapore, the Philippines, and Southeast Asia by Alkem Company (S) Pte., Ltd., Singapore; in the English language in India, Bangladesh, Sri Lanka, Nepal, and Mauritius by Prism Books Pvt., Ltd., Bangalore, India.

Printed in the United States of America

Library of Congress Catalog Number: 2004016872

ISBN-10: 0-932633-59-5
ISBN-13: 978-0-932633-59-0

12 11 10 9 8 7 6 5 4 3

HIRING THE BEST KNOWLEDGE WORKERS, TECHIES & NERDS

◆ ◆ ◆

The Secrets & Science of Hiring Technical People

by Johanna Rothman

foreword by Gerald M. Weinberg

DH
Dorset House Publishing
353 West 12th Street
New York, New York 10014

Dedication and Acknowledgments

I dedicate this book to my family—Mark, Shaina, and Naomi—for whom I shall be forever thankful.

I am grateful as well to many friends and colleagues for their review of early drafts of my manuscript. Any mistakes that remain are mine! In addition to the editorial staff at Dorset House Publishing, I thank Nicole Bianco, Esther Derby, Dale Emery, Paul English, Sally Hehir, Erik Hemdal, Elisabeth Hendrickson, Cem Kaner, Bob Lee, Vijay Manwani, Jonathan Ostrowsky, Janna Patee, Bret Pettichord, Dwayne Phillips, Barbara Purchia, Rob Purser, Stever Robbins, and Sally Silver for your varied and invaluable comments—each of you have contributed in ways I could never have expected. I thank you all.

Click here to toggle this file's Bookmarks pane.

Contents

Illustrations

Foreword

I've been consulting with high-tech firms for half a century, and I sure wish I had this book fifty years ago. I cannot even estimate the number of times I've seen hiring problems that would have been prevented by a manager reading *Hiring the Best Knowledge Workers, Techies & Nerds.* At least once in every consulting assignment, a manager asks me a question that could be answered by a quick look-up in Johanna Rothman's contents or index. I know, because many times in the past decade, I've recommended Johanna as a hiring consultant to my own clients, and she's never failed to produce phenomenal results.

Hiring mistakes cost high-tech organizations literally billions of dollars each year, plus untold pain and anguish on the part of hiring managers, applicants, and employees. This superbly organized book distills Johanna's many years of experience doing just what its title implies—Hiring the Best. It nicely balances cases drawn from that experience with principles abstracted from dozens or hundreds of cases. Indeed, it would be a poor manager who couldn't pay for his or her own salary by applying these principles to an organization's hiring process.

There's only one thing I can find wrong with this great book: The author underestimates its value. She wrote it as a book for managers who are hiring, but that's too narrow an audience. It excludes several other large groups whose members should be reading it:

1. Everyone who participates in the hiring process, such as coworkers who are called upon to interview job candidates
2. Teachers and trainers who prepare students for jobs and need to understand the processes that will be, or should be, used to select them

3. Any knowledge worker, techie, or nerd who is now, or will be in the future, applying for a job, or even for a promotion in his or her present job

But, in the end, *Hiring the Best Knowledge Workers, Techies & Nerds* is a book for managers, and any high-tech manager who doesn't read it as a hiring manager may soon be reading it as an out-of-work applicant looking for a new job.

July 2004
Albuquerque, New Mexico

Gerald M. Weinberg
Author of *The Secrets of Consulting*
and *The Psychology of Computer Programming*

Preface

I’ve had the opportunity to hire or participate in the hiring of hundreds of technical people over the years, including developers, testers, technical writers, technical support staff, pre- and post-sales applications engineers, consultants, leads, and their managers. I’ve been part of interview teams charged with hiring product managers, electrical engineers, mechanical engineers, in-house teaching staff, and information systems staff.

Hiring technical people has never been easy. Many organizations persist in a near-constant state of having too few *qualified* technical workers. When the economy is strong, we attribute the shortage to too few qualified candidates to fill the many openings. When the economy is weak, we attribute the shortage to too many poorly trained applicants with unsuitable backgrounds.

Too often, we apply the same hiring techniques to knowledge workers that we use to hire skill-based staff. Skill-based staff members possess a set of tools and techniques that can be applied in the same way in almost all situations. Technical people—in particular, knowledge workers—must adapt their knowledge to the specific situation. Such workers are not just the sum of their technical knowledge; they are the sum of both *what they know* and *how they apply that knowledge* to the product. In particular, how they use their technical skills to benefit the product, how they manage their work, and how they manage their relationships with other people all must be assessed when hiring and evaluating a knowledge worker.

While there are some similarities in the hiring process, hiring technical people—knowledge workers—is vastly different from hiring purely skill-based staff.

Knowledge workers have unique qualities, preferences, and skills—such workers are not fungible assets.[1] The ability to adapt knowledge and to innovate makes one developer, tester, project manager, or technical manager different from another. That difference among people is key to making good hiring decisions.

You want your organization to succeed, and so you need to know how to define and assess a technical candidate's qualities, preferences, and skills, but you also need to be able to predict a technical person's chance at succeeding in *your organization*. The techniques and recommendations set forth in this book are designed to make hiring a streamlined, efficient, and satisfying experience.

Why read this book?

"The whole interviewing thing takes forever."

"How do I know this candidate will work out?"

"I can't seem to find candidates who meet the job's specifications."

I hear comments like the preceding every day. The comments express the frustration that many technical managers feel as they attempt the very difficult job of hiring. If you're like most of the technical managers I've worked with, you may not be sure how to define the job's requirements, how to find suitable candidates, what skills you need to interview well, or how to make an offer that the candidate will accept. Most technical managers who ask me for guidance in hiring technical staff find it difficult to define appropriate requirements, to assess experience and cultural fit, and to check references in a way that makes sense for the candidate and themselves.

Or, possibly, you know how to do all of that, but the hiring process consumes more of your time than you comfortably can allocate. If you have any of these problems, the material I present in this book can help you.

Many books on hiring give good advice on how to ask questions in an interview or help you develop reference checklists. But few books address how a technical hiring manager can create an efficient and effective hiring process for a technical organization. If you hire technical people, you know that the approach used to interview, assess, and hire skill-based staff does not do the job when you're trying to hire knowledge workers or to evaluate the skills of those technical people already in your group. You need to precisely define the specific experience, qualities, preferences, and

[1] For a detailed exploration of "fungible assets," see Tom DeMarco, *Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency* (New York: Broadway Books, 2001), pp. 13-21.

skills you want people to have, and you'll need a specific strategy to help you detect whether a particular candidate has the necessary expertise.

Here's what I hope this book can do for every hiring manager who uses it as I've intended it:

- Save you time and money every time you hire.
- Help you hire people who can perform the required work well.
- Help you screen, evaluate, and hire the right staff for your specific organization.
- Eliminate the wasted time and suffering that result from having to fire people who should not have been hired in the first place.
- Help you develop and demonstrate fundamental management competency.[2]

Save time and money. This book offers a streamlined approach to hiring. The more you streamline your hiring approach, the faster you will be able to evaluate suitable candidates, and the better the hiring decisions you'll make. An effective hiring process is especially important when you consider the toll a bad hire can take on your organization. Add up the direct monetary costs of recruiting, the cost of the time you and your staff spend on hiring the person, and the actual cost of the person's salary and benefits while he or she works for you, and you'll quickly see that the cost of a bad hire can be enormous. Perhaps equally costly, a bad hire saps energy from the work your organization is trying to perform, and prevents the work from moving forward. A bad hire doing substandard work can even damage your product to such an extent that it will have to be redeveloped from scratch.

Hire people who can perform the work. Few things are worse than feeling that a new employee somehow misled you on his or her résumé or in the interview. You thought you hired Dr. Jekyll, but Mr. Hyde showed up to work. Or, you thought you hired "the best and the brightest," but the people who came to work seem mediocre and dim. If you use the job analysis template in this book to help you define the qualities, preferences, and skills needed for a particular job, you can create a set of strong interview procedures that should prevent you from simply hiring just a warm body.

[2] For a solid treatment of the hiring manager's role in finding talented staff members to hire, see Ed Michaels, Helen Handfield-Jones, and Beth Axelrod, *The War for Talent* (Boston: Harvard Business School Press, 2001).

Screen, evaluate, and hire the right staff for your specific organization. I've learned from my early hiring mistakes and know ways to avoid making those mistakes again. There *are* good people out there, no matter the state of the economy. You can help yourself hire well by first defining a standard for what "a good employee" is for your specific organization, and then translating that standard into precise job requirements, a sound job description, and a comprehensive listing of information needed for successful interviewing.

Fire fewer people. Most managers dislike the firing process: the warnings, the get-well plan, the actual firing. Many ignore the problem altogether or shunt the non-performing employee to other projects or other managers. If you would like to avoid the firing problem, use this book to help find people who can perform the work.

Develop and demonstrate your own management competence. If you'd like to be a better manager or you'd like to advance in management, this book can help you hire well. Simply put, managers who can tell the Jekylls from the Hydes will be more successful than managers who cannot. Likewise, a manager with a staff whose members can work with others in the company will be able to complete his or her assignments faster and with greater success.

The bottom line: Once you've defined what a "good employee" means for your needs and your culture, you can quickly review résumés, conduct interviews, make offers, and hire the right technical person for the job. I hope the numerous tips, suggestions, and recommendations in this book will help you expedite the hiring process as well as make it a more pleasurable experience.

June 2004 J.R.
Arlington, Massachusetts

HIRING THE BEST

KNOWLEDGE WORKERS, TECHIES & NERDS

Part 1

Defining Requirements for Yourself and Your Candidates

The three chapters in Part 1 deal with defining requirements: developing the hiring strategy, analyzing the job, and writing the job description.

The hiring process can be viewed as a series of steps, each of which requires some preparation. The schedule I recommend you use for successful hiring follows, showing how much time I generally allocate for each step.

From beginning to end, the total time per candidate should be approximately one day plus the time you'll spend on sourcing activities and résumé review. If you're spending more than one day on each candidate, review your preparation work. Don't waste time on things you can preplan or organize. Define the job so everyone who's recruiting on your behalf understands the open position. In addition, if you're spending more than a few weeks recruiting candidates, reorganize your recruiting mechanisms to recruit more effectively.

Preparation (Time: 2 hours per open job)

Step 1. Define the requirements for the job:

a. Define your hiring strategy, identifying why you're looking for people (30 minutes).
b. Analyze the job, detailing the requirements a person should satisfy to be successful (30 minutes).
c. Write the job description, describing what interviewers should look for when evaluating candidates (30 minutes).

Step 2. Write and place the job advertisement (30 minutes).

Sourcing (Time: 3 hours per candidate)

Step 1. Select your sourcing mechanism—that is, the techniques you'll use to attract suitable candidates. Work with your HR staff to implement those techniques (10 minutes, plus actual recruiting time if you attend job fairs or other networking events).
Step 2. Filter résumés, reviewing each to determine whether you want to phone-screen the candidate (30 seconds per résumé).

Interviewing (Time: 3 hours per candidate)

Step 1. Define the list of questions you'll ask each candidate on the phone to qualify him or her for an in-person interview (10 minutes).
Step 2. Phone-screen each candidate, conducting a brief phone interview to determine whether you want to interview the candidate in person (10-45 minutes).
Step 3. Schedule the in-person interviews, selecting a team of interviewers and planning who will ask which questions when (60 minutes).
Step 4. Hold a follow-up meeting with interview-team members to hear their perspective on the candidate (15 minutes per candidate).

Offers (Time: 2 hours per offer)

Step 1. Check references (60 minutes).
Step 2. Extend the offer (60 minutes).

Before you read further, I must issue my own requirement: Some of the advice in this book may conflict with your organization's hiring policies and practices. If at any time you're not sure whether something I've suggested is appropriate for your organization, your corporate culture, or even your geographic location, check with professionals in your company's Human Resources or Personnel Department, or with your corporate lawyer. They will factor in conditions specific to your particular situation; above all else, *follow their advice.*

1

Developing Your Hiring Strategy

> *"I wish I'd never hired Zeus. I know he's excellent at his technical work, but he's so difficult. He intimidates people, hurling words when he's displeased. What a mess."*
>
> —Statement from a dissatisfied manager

Hiring "messes" do happen—but they are something we can and must work to avoid. In truth, the decision to hire an employee is one of the most critical decisions a manager can make. You, your team, and the organization will live with the long-term consequences of your hiring decision. Getting it right the first time can be challenging but, with a good bit of hard work and intelligent planning, it can be done. When you invest your time developing a hiring strategy that defines the kinds of people you need and that helps you determine how such people will fit into your organization, you can improve your success ratio dramatically, perhaps even to the point of achieving a perfect record. A well-constructed strategy should pay big dividends for the future of your group.[1] One good way to start is by asking questions such as the following:

1. What kinds of people are you looking for?
2. Which roles do you want to fill first?
3. What talents or skills are most important to your search?
4. How will you decide on a candidate?
5. What will you do if you can't find the right people?
6. How do you know when you've got the right person?

[1] For guidance on linking your corporate strategy to hiring, see Ed Michaels, Helen Handfield-Jones, and Beth Axelrod, *The War for Talent* (Boston: Harvard Business School Press, 2001).

To help find answers to these questions, let's look at some real-world scenarios, tales taken from the workplace, possibly even from a workplace you'll recognize.

Kent, a test manager, has a great test group. He needs to hire someone else, and his first thought is, "I need someone just like Louise; she found all those bugs in the last release." *Should Kent be looking for other testing skills?*

Sharon, a development manager, suspects that her group's architecture skills are weak, but because the people in her group all work well together, she doesn't know what to do. *Should she hire the senior architect who might not get along well with the rest of the group, or someone who can fit into the group well? Does she have to choose between the two options?*

Dan, a hiring manager, just fired his fourth new hire in three years—an unusually high percentage of the ten people he's hired in the same time period. Even though this most recent ex-employee's technical skills were appropriate to the job, the employee consistently infuriated and alienated his coworkers. *Is Dan hiring the best-suited people for the culture?*

Beth, a technical support manager, works for a company that is changing its product line from an expert-use-only product to a suite of products that will be used by a range of people with various levels of expertise. Her staff is great at supporting Ph.D.s, but can its members support administrators and lab technicians also? *Should Beth be hiring from a different skill-set?*

Each of the four tales depicts a common hiring problem, possibly one you've experienced firsthand, but I'm not going to tell you what I think Kent or Sharon or Dan or Beth should do. Instead, I'll arm you with the questions I asked them to consider, questions you, too, can use to develop your hiring strategy:

- *Should you hire someone just like yourself or just like the rest of your staff?* Sometimes you may want to hire someone who will fit easily into the team. At other times, you may want to hire someone completely different, so that you can take advantage of differences in experience, skills, and personal qualities.
- *What are your tradeoffs in your hiring?* Do you have only one requisition authorizing you to hire one candidate, but you need two people? If you have to make tradeoffs, you'll not only have to decide which aspects of the job you're willing to sacrifice, you'll also want to make sure that the candidate can perform the most important work successfully.

- *Are you looking either to overcome certain weaknesses or to expand certain skills in your group?* Kent, our test manager, may decide to bring in someone with a wide range of experience using automated testing tools in order to try different kinds of testing for the product. Or he may decide that another exploratory tester has just the right mix of skills.
- *Are you hiring based on a candidate's specific technical skills, but firing people because they can't fit into your culture?* If you find yourself having to fire people within a year of their hiring, the reason may be that you have trouble assessing how a candidate will fit into your group or into the larger organization.
- *What kind of diversity are you trying to achieve in your group?* Having only senior-level developers or manual testers in a group assures that the group will lack skill diversity. Staffing with only men or only women indicates another lack of diversity. Having only extroverts or only introverts is yet another form. Maybe the members of your group know everything there is to know about using the development tools but not how the customers actually use the system. When you consider diversity, don't just look at the obvious; look at *all* the issues associated with skill level, experience, and how people communicate.[2]
- *Are you looking for a personality who will complement and expand the capabilities of the current team, or are you looking for someone who will fit into the team seamlessly?* If you have a group that already solves problems well as a unit, maybe you should hire someone who fits into that group. However, even a group whose members work well together can use some jiggling every so often, if only to spark new ideas.

A well-thought-out hiring strategy will help you identify the candidates you should interview.

Ask questions when creating a hiring strategy.

Before you determine your hiring strategy, there are several important questions to consider:

[2] Diversity can be enhanced through your hiring strategy. For sage advice on other factors to consider than just a candidate's qualifications, see Ed Michaels, Helen Handfield-Jones, and Beth Axelrod, *The War for Talent* (Boston: Harvard Business School Press, 2001), pp. 13ff.

- What problem(s) are you trying to address with the hiring you're planning, and do you have the facts you need to assess your current team's qualities, preferences, and skills?
- Which roles are most important to fill, and in what sequence?
- What are your criteria for choosing suitable candidates?
- What is your decision-making process?
- What contingency or risk-mitigation plan will you use if you can't find the right people to hire?

Let's start by considering the problem you want to solve with your hiring.

Identify the problems you should address.

Some hiring managers begin the hiring process by compiling a wish list of technical skills (two years of C++, four years of Java, five years of UNIX, three years of project management, and so on). Unfortunately, a detailed skill listing tends to constrain the position, so that no candidate perfectly fits the role.

Instead of focusing on technical skills, think of hiring as a problem-solving exercise. I focus on five problem-solving steps:

- Define the problem you hope to solve by hiring this person or these people.
- Develop a strategy for identifying the candidates who are best-suited to your needs.
- Assess your current staff to see where you need complementary skills or experience.
- Define the kinds of people you require.
- Hire the person or people you need to solve the problem.

In the following paragraphs, I describe common problems and some ways to handle them. Some problems may resemble ones you have been trying to solve.

You need additional people to do more of the same kinds of projects that are currently being done. If you need to hire more people to do the same kinds of projects, put extra emphasis on each candidate's technical skills, *if you can find enough candidates*. If you can't find technically skilled candidates, then look for candidates who fit into the existing culture or who have demonstrated an ability to adapt and learn—and plan to train your new hires. Consider whether you need junior-level people, senior-level people, or people who are experienced technical leaders. Especially when you're hiring a lot of people at one time, make sure you don't base your decision to eliminate candidates solely on the fact that they seem to have too little or too much experience. Junior-level candidates can grow along with your organization, and can be the leaders in a few years. Senior-level candidates can bring significant problem-solving expertise into your organization.

Your work is changing focus from one kind of work to another. If project staff members must make the transition to a different kind of work, you may need to add people who are different from those currently on your staff. One test manager recently told me, "My folks are great at testing the product from a black-box perspective. However, that's all we do. With this new product, I need to modify the testing to include performance and reliability testing, something people on my staff know nothing about. They just don't have the technical background to know how to perform this kind of testing." This problem is especially challenging if you cannot add staff, but must lose current staff to make room for people who are qualified to perform the new assignments. In this case, you'll want to pay special attention to the required functional skills for the job.

Your technology is changing from one technology to another. When you're recruiting because your company must make the transition to another technology, consider a candidate's problem-solving skills, adaptability, and cultural fit for the new organization, rather than focus on the person's current technical skills, particularly since you'll need to train your staff in new skills anyway. For example, if staff members in the new technology use a different programming language from the language used in the old environment, it's easier to assess suitability among candidates who've already learned multiple languages, rather than selecting people who have worked in only one language.

Decide how many people you need, at both junior and senior levels, and in which levels new staff need to possess technical expertise. One option to consider when moving to a new technology is whether to hire someone with significant expertise in the new technology to mentor both your current staff and new hires. Keep in mind that in order for this technical mentoring to be successful, the expert will need to build a rapport with the team quickly.

If you're adding new technology and still supporting the old technology, don't hire a new team to work on the new products and keep the existing staff working on the old products. The existing team may want to work with the new technology and may become frustrated that new people will have all the fun work. If you break the work up into new (read, "exciting") and old (read, "boring," "tech. support," "housekeeping") work, you'll create more problems than you solve. If you want to retain your current staff members, ask them what work they want, and hire to backfill their current roles so they can move on to do the new work.

You're on the cutting edge of technology but you're so far ahead of the market that you don't know what technical skills the staff should have. Sometimes, when you're on the cutting edge, you may find it difficult to determine precisely which technical skills will be needed. Here, a good strategy is to place emphasis on a candidate's adaptability, cultural fit, and ability to work in teams. Consider the experience level and the technical leadership abilities of the candidates. Think in terms of what work, call it "X," must be done in your cutting-edge project. That way, even if you don't know the specific required skills, you can ask candidates to describe their experience doing "X."

Years ago, before configuration management systems were common, I needed to hire a release engineer, someone with expertise in builds, branches, and what we now call configuration management. Since I didn't know precisely what skills would be required, I looked for a candidate who could communicate well with system developers, and who had demonstrated an ability to organize complicated work and run smoke tests. I suspected I didn't need someone with years of experience, just an exceptionally good problem-solver. The candidate I hired had only two years of experience, but he'd worked as a programmer throughout high school and college. He was a great release engineer, and now is a highly qualified configuration manager.

You're putting together a brand-new group or are adding staff to a recently established group, but you don't know what personalities, characteristics, and skills will make the best mix. If you have a newly formed group or are adding people to a group that has not been together very long, the people you add should enhance the group's ability to work together and mesh; they should not prevent group members from working well together. Hiring a personality who doesn't fit well will prevent your team from doing the work. An established group, whose members are confident of their abilities, can handle different personalities and challenges to the current work; new groups are less likely to succeed.

In order to be successful, a new group needs to build confidence and develop ways for individual members to work together. You'll want the most experienced people you can afford, because you need people who can manage their work while developing healthy, working relationships with coworkers. For this group, you need people who are experienced in both technical and communications skills. As your group matures, you can hire less experienced staff.

You need to change what your group can deliver, but you don't know what skills to add. Maybe you have a great group of developers and it's time to add some testers. Maybe you've got writers, and you need some editors. Maybe you've got manual testers, and it's time to add some automation to the mix. Whatever the case, if you're looking for people to fill a gap, you'll want to consider functional skills, but don't forget to assess each candidate's cultural fit. Because new people already will be rocking the boat just by virtue of performing a new function, they particularly need to fit in with the team and the culture.

Adding people with different skills to your team tests its maturity and adaptability. Your challenge is to overcome the second-round effect in which new people join an established team but are not perceived as full partners. Look for people who, in addition to possessing superb technical skills, can quickly learn and adapt to the team's culture.

I once worked with a manager who brought ten new people into what had been a four-person group. He had hired the new members on the basis of specific skills (user-interface development, testing, and so on), but he did not consider how well they would fit into the existing team. After sixteen months, it was obvious, even to an outsider, who the original members were and who'd been hired later. The fourteen-member team couldn't make the project succeed until both the original team and the new members changed their behavior and adapted to each other. If the manager either had hired more adaptable people for the original group or had focused his second-round hiring on people with better communications skills, the team would have meshed much sooner.

Your group must finish a project faster than originally planned, but you don't know whether adding staff will increase productivity. The good news is that you've got a group of people who work well together, but you need to increase productivity. Sometimes, adding people to a team is the answer to attaining a faster release, but bringing them up to speed may counteract the contribution that additional staff should eventually make. If productivity can be increased by assigning new people to work on parallel projects or if new members can work in parallel with the original staff on one project, and management can handle such a challenge, by all means add staff. If you make the decision to add staff, bringing in candidates who fit your culture is critical.

You need a few additional people right now, but they won't be needed forever. Sometimes, you need people *right now* on a project, but you don't want to keep them in the company long-term. If this situation is likely to arise, you might choose to hire fewer permanent staff members and more contractors. Analyze your immediate, mid-term, and long-term needs to decide which of your candidates should be offered a contract and which should be offered employment.

When you hire contractors, consider their communications skills to be equally as important as their technical skills. If you don't intend to keep someone around for a long time, you'll need that person to be conscientious about thoroughly documenting what he or she does, detailing how all aspects of the job are performed. Make sure that when the contract term expires, the contractor is capable of handing off the work to other people. For the most part, I interview contractors the same way I interview permanent staff, although I do give greater emphasis to such areas as their ability to complete and hand off work, and their communications skills. I treat this important topic in greater detail in Chapter 7, "Developing Interview Questions and Techniques."

You have to fire more of the people you hire than seems reasonable, but you aren't sure how to get a more stable staff. If you find that many of your new employees are not successful at their jobs, or that you fire even 5 percent of your new hires, reassess both the content of your interview questions and how you or your interviewers ask those questions. The most effective screening involves behavior-description questions that include some combination of technical-skill and cultural-fit analysis. Do you and your interviewers have enough technical and interviewing expertise to ask the appropriate technical questions and assess the answers? If not, you'll need to change interviewers, and increase the interview team's level of expertise.

If you have to repeatedly fire people because their technical expertise is inadequate, you're probably not asking specific-enough interview questions. If you have to fire people because they don't fit into your group, perhaps you haven't fully identified the kinds of people that best fit your culture.[3] If you have thoroughly defined the kinds of people you need and you still must fire people because they don't fit into the organization, you're probably not using a consensus-based approach to candidate appraisal. You'll discover more about the candidate's qualifications and his or her fit with your culture if you invite several people outside of your

[3] For a sound discussion of hiring and firing complexities, see Jim Harris and Joan Brannick, *Finding & Keeping Great Employees* (New York: AMA Publications, 1999), pp. 18ff.

group to be part of your interview team. Generally, when you use a consensus-based approach to candidate appraisals, you develop more of an understanding about what your team wants. This topic will be treated in more depth in Chapter 10, "Following Up After the Interview."

A True Story

Fred is a non-technical, quick-to-judge MIS manager. When I first talked to him, he boasted that he could interview someone and know within thirty seconds whether the person would fit into the organization. I didn't hear from Fred again until after he'd fired two people before their three-month anniversary with the company. More than a little bit rattled, Fred decided to ask for my help with interviewing and the hiring process. I suggested that Fred recruit people from the rest of his company to help him with the interviewing, and gave him guidelines to follow. He assembled a group fairly quickly that included an MIS technical staff member, the release engineer, the support manager, and a couple of developers—all people who understood the implications of MIS work and who possessed some of the expertise required for the jobs to be filled. Once assembled, the interview team posed questions they wanted to ask, and Fred agreed to let them go ahead with the interviews. Fred also agreed to withhold rushing to judgment during each interview. Following each interview, the interview team met to discuss the qualifications and suitability of each candidate. Happily, this approach enabled the team to find two candidates to replace the fired employees, producing an MIS group that remains stable and successful to this day. At last check, the new employees were still working at the company, almost three years later.

You have too much turnover, but you don't know how to reverse the trend. If your employees choose to leave after they've worked at your company for only a year or two, maybe they were not the right candidates to hire in the first place. Unless you have defined the job as an entry-level, short-term position, you do not want people to view your company as a temporary port in a storm. Such employees are not good long-term investments and are not worth the training you'll undoubtedly need to invest in them. If you find yourself facing excessive turnover, you're probably not asking the right interview questions.

A test manager stated that she'd replaced five members of her ten-person group in one six-month period. As this turnover rate was unusually high, I recommended that she contact the former employees to retroactively conduct exit interviews in order to find out why so many people had left. The exit interviews gave her the reason: She learned she had consistently hired people who were risk-takers who enjoyed solving problems in unique ways. The development organization didn't value those testers, and wanted to work with testers who planned testing in a predictable way. The hiring problem wasn't that she was hiring people with poor technical skills; the problem was that she hadn't given enough thought to the cultural-fit problem: how to hire risk-taking testers who had enough patience to continue working through the changes she was trying to implement. By changing her cultural-fit questions to identify how the testers tested, and by looking for people with patience for cultural change, she was able to keep turnover to a minimum.

Your ability to recruit more people to join an existing project has become increasingly difficult, but nothing you do seems to change the picture. If finding people is difficult, maybe you're not using enough different approaches or recruiting mechanisms. If you use only one recruiting method, you run the risk of missing out on potential candidates. For example, if you only use classified ads, you'll miss people who only work with recruiters. If you only use one general-purpose, Web-based job board, you'll miss people who use industry-specific sites or geographic sites.

For more details on how to build and use your recruiting network, see Chapter 4, "Sourcing Candidates." Also, check to see that you're not inadvertently discriminating against people who are different from you. Chapter 6, "Reviewing Résumés," describes ways to check for your prejudices.

You have a solid group of people on the project, but everyone has the same set of skills or philosophy of work. You'd like to add more diversity. Sometimes, when a group has been together a long time, its members may start to think alike. The best remedy for this is to shake up the group by gradually adding people with different personality types or backgrounds. If you're changing the focus of your product base, you might add people who more closely reflect your customer base. Or, if the team consists of people primarily of one gender, race, or philosophical outlook, hiring people of the other gender, from another race, or committed to other philosophies will make for a richer work environment.

In technical groups, technical skill and expertise are usually valued more highly than personality or race or gender. That's not to say there are no bigots or prejudiced individuals in technical groups, but that, in my experience, most people are more interested in what another person can do than in what the person looks like or how forceful his or her personality is. Not surprisingly, we tend to neglect considering personality diversity while hiring. How a person solves a problem or performs an assignment is influenced by his or her personality, and it can be used to advantage in matching a candidate to a job. For example, many people working in the technical field are quick to make decisions, but creative product architects may choose to ponder several designs, looking at the pros and cons before coming to a conclusion. Some testers like to plan their work; others explore a less structured path as opportunities arise. Some people prefer to talk out the issues; others prefer to think about the issues privately and then discuss them.

Look at the range of personalities on the existing team to see whether all team members have one kind of personality. The more diversity you have in personality types, the less likely you are to be blind-sided by a problem no one considered.

Sometimes, diversity can be achieved by mixing experienced workers with entry- or junior-level staff members. Such a mix would have benefited one development manager who told me, "Everyone in my group has at least ten years of valuable experience. Most are designers, but we also have three real architects. Unfortunately, I don't have enough high-level work to keep them all busy right now. I need junior people to be my senior folks' journeymen, so I have a more natural mix of engineers."

You can hire junior-level people to perform jobs that do not require senior-level knowledge and talent. Allow for the maturity hierarchy of technology skill and knowledge to take a natural path—mix experience and knowledge levels.

You need more management capability, but no one you've interviewed is qualified. If your group has grown in size, or if you have a start-up group that must make the transition to the next level to become a more productive entity, you may need to hire more managers. The chief technology officer of a Web-based start-up typically might manage a technical group of twenty software developers, testers, and operations staff members for years with only the help of technical leads in the various functional areas, but if the organization decides to hire another five people, then experienced, full-time managers, not just technical leads, will be needed to make sure all of the management tasks will be accomplished correctly, on time, and within budget. In Chapter 14, "Hiring Technical Managers," I provide detailed information about hiring managers.

Sometimes—most times, in fact—you have *more than one problem to solve.* When that happens, list all your problems in order of their priority so you can then determine whether you need to hire additional staff members based on technical skills, cultural fit, adaptability, technical leadership, or some other quality, preference, or skill. Once you've defined each problem and determined its importance in relation to the other problems that also must be solved, you can choose which types of people you need to hire first. This topic is discussed more fully in the following section.

Determine which roles you want to fill first.

When you're hiring more than one person, or hiring into a group over time, decide which capabilities are your highest priorities. Not all roles in your organization are the same. If you need a product architect, then a designer will not do. Whether you're hiring for a product company or for an information services group, consider the kinds of people you want and your priorities.

You will need to make decisions as you build your list of first, second, third, and so on, hires. Some typical tasks are identified in Table 1-1, below, which suggests job titles to fill function areas:

Function to Be Performed	**Possible Roles and Job Titles**
Requirements Analysis	systems engineer, analyst, requirements specialist
Development	systems architect, senior designer, junior designer, programmer, project manager, technical lead
Release Engineering	build engineer, librarian, configuration manager, operations analyst
Testing and QA	automated tester, manual tester, exploratory tester, test project manager, technical lead, metrics gatherer
Documentation	editor, writer, book designer, technical lead, production specialist, graphics artist
Support	tier-1, -2, or -3 support (first-line, mid-level, and senior-level support staff)
Usability Engineering	interaction specialist, designer
Project Management	project administrator, project manager, program manager

Table 1-1: Function/Role Chart.

A True Story

> A chief technical officer of a Web start-up defined his current hiring needs: “We’ve moved past the initial start-up phase. We have three developers—I guess I’d call them senior designers—and I’ve been doing the architecture. It’s time to bring in a project manager and some testers, so we can ‘product-ize’ this beast now that we’ve got the funding. But, these people have to work *with* us, not against us. I’m not ready for formal release engineering, or formal process definition, or formal system tests, but I am ready to start automating tests of the product core. We need a technical project manager, an automation tester, and one more tester who can find the problems we developers don’t see.”

The CTO in the preceding story is trying to solve the staffing problem by filling in positions with other skills. Since his group is small, he’s considering cultural fit (qualifications he says he is “not ready for”), but the driving force behind his hiring is to bring more people on-board to do different work than is done by the people he already has.

Once you’ve decided which roles you want to fill first, go back to assess your current staff members and the roles they perform, making sure they continue to fill those roles appropriately. If they still perform jobs that partially fill the problem areas you’re trying to staff, include them on your interview team. When the people already fulfilling some of these roles participate on the interview team, you can obtain a richer picture of each candidate. If these current staff members can no longer perform the jobs you need done, determine how many of which kinds of new people to hire, and decide how you’re going to manage the problem of your current staff’s inability to perform the needed work. For more guidance, see Chapter 15, “Moving Forward.”

If your hiring strategy includes hiring many people at one time, you may be lucky enough to find candidates for positions you need to fill but weren’t planning on staffing until later. If this happens and you have the budget to support these additional employees now, hire them! Then, replan the work your group will do, and update your hiring strategy.

Decide which criteria matter most.

If one person could make all the hiring decisions, the hiring process could be accomplished comparatively quickly. However, in most organizations except possibly the smallest, one person probably can't ask all the questions and assess how each candidate would respond in different situations and to different people. To facilitate the process, create an interview team so that multiple people can interview candidates. Ideally, the same individuals you select for the interview team will be available to interview *all* the candidates. Once all members of the interview team have assessed each candidate's technical qualifications, proficiency level, cultural fit, and so on, invite interview-team members to compare notes on each candidate, encouraging them to share both positive reactions and negative impressions. Assure people on the interview team that their comments will be kept completely confidential. In some circumstances, it is advisable to require all members of the interview team to sign a simply worded confidentiality agreement prior to beginning discussion. Clearly, interview-team members must trust each other *and you* sufficiently to open up freely, but a confidentiality agreement can help to emphasize the seriousness of their involvement in the hiring process.

Train your interview team to apply a limited-consensus approach to hiring. When groups use limited consensus, not everyone may agree with the decision, but each person should be satisfied enough with a particular candidate's suitability not to block the decision to hire him or her. Limited-consensus discussion sessions provide the following benefits:

- Interview-team members feel valued because they are included in the decision-making process, which builds camaraderie and makes them more likely to help integrate the new staff member into the group.
- Interview-team members develop a stronger sense of their organization's culture, and learn how to successfully use it.
- The manager learns about the candidate from the different perspectives of each of the interview-team members.

The manager ultimately is still responsible for the hiring decision, but he or she doesn't have to gather and evaluate all of the data about each candidate single-handedly. For more discussion of how to appropriately use multiple interviewers, see Chapter 9, "Planning and Conducting the In-Person Interview," and Chapter 10, "Following Up After the Interview."

A True Story

Steve, a development manager, described his interview team this way: "Our HR guy talks to each candidate to assess whether the person will fit into this small-company culture. Then, a senior developer asks difficult questions about the kinds of products this candidate has designed and how the candidate makes design decisions. The test lead then talks to the person about how he or she manages, develops, and tests code. One of the mid-level developers is great at developing audition questions, so he runs the audition part of the interview. The hiring manager comes at the end, following up with more problem-solving and cultural-fit questions to determine whether this candidate will fit into the group."

As Steve has described in the preceding True Story, everyone in the group has a role—and an audition-style format is integral to the process. Each interviewer takes a different aspect to investigate in each candidate. When Steve and his group members put all their observations together, sharing various perspectives, they have a clearer picture of the "whole" candidate. If someone sees a candidate as assertive during the interview, and another person sees that assertiveness as bossiness, the two interviewers can compare their different perceptions. For more on auditioning techniques, see Chapter 7, "Developing Interview Questions and Techniques."

Steve's group uses a limited-consensus, decision-making process. Since team members have all seen a different aspect of the candidate in the interview, they can now discuss the candidate and share their conclusions about the candidate.

A team of people is helpful for interviewing a candidate, and it makes sense to use that team both to evaluate the candidate and to help decide whether to hire the candidate.[4] With enough people on the interview team to contribute multiple perspectives, the hiring manager's job becomes easier. If you worry that too many perspectives may make it difficult or impossible to make a decision, read Chapter 10, "Following Up After the Interview." For more information about audition techniques, see Chapter 9, "Planning and Conducting the In-Person Interview."

[4] The topic of using a team for the process of hiring is nicely treated in Cem Kaner, James Bach, and Bret Pettichord, *Lessons Learned in Software Testing: A Context-Driven Approach* (New York: John Wiley & Sons, 2002), pp. 206ff.

Identify what process you'll use in decision-making.

Consensus-based hiring provides a great way to manage the potential risk associated with making a bad hiring decision. Here's what happened when one hiring manager, an SQA director, ignored the concerns of her group: "I desperately needed a release engineer. A friend of mine was available. I knew she could do the job, and I wanted to hire her outright. When she came in for the interview, half the people on my staff said she wasn't going to work out. I was desperate, so I hired her anyway, but as a short-term contractor. It's a good thing she was a contractor and not a permanent employee, because after she'd been here about seven weeks, she had made enemies out of almost everyone. People refused to talk to her and were on the verge of refusing to work with her. If she'd been a permanent employee, I would have had to fire her. The fact that she was a contractor allowed me to sever the work relationship without destroying our friendship. I will never ignore what my group says about a candidate again."

Decide who in your organization will help you make the hiring decisions. If a valued employee is voluntarily leaving, consider using that employee to evaluate potential candidates. I've had good results using such people to help interview their replacements.

Some managers worry about consensus-based hiring decisions, arguing, "Doesn't consensus-based hiring allow one employee to veto the entire process? What if the vetoer is someone I don't want to keep in the organization?" The answer to this is simple: In the interview process, only involve employees whose work you respect and value. If an employee isn't successful in his or her technical position, don't make that employee part of the interview team.

When you have an interviewer who consistently vetoes candidates, do the following:

1. *Make sure everyone realizes why you're hiring another person.* Determine whether anyone has anything to fear from this new hire. Deal with that fear now, before you begin the interview process. Otherwise, you won't be able to hire anyone.
2. *Clarify the job description.* Sometimes, people veto candidates because they don't understand the role you want filled in your organization. For more both on defining the problem you're solving and on clarifying the job description for inter-

viewers, see Chapter 9, "Planning and Conducting the In-Person Interview."

3. *When someone vetoes, ask why.* Drill down to discover the reasons, and check those reasons with other interviewers.

I learned of the following stalemate from a development manager in a start-up organization: "As development manager, I tried to hire a second developer into the group. The first interview day, we interviewed three people. My on-staff developer's first reaction was, 'Not one of these guys is any good. Time to look for more people.' I was surprised, and said so. I asked why he thought none of the candidates was any good, because I'd thought two were great, and one was okay. He said, 'They'll never make it here.' I asked why. He said, 'Because they're not as smart as I am.' I asked if that was a requirement for him, because it wasn't a requirement for me. He said it was, stating, 'With only two of us, I need to make sure I can learn from the other person in the group. Otherwise, I'm going to be carrying that person.' I asked whether any candidates would be acceptable if we changed our technical mentoring. 'Oh yes, the first guy was great.' My developer was trying to make sure that both developers shared the work and the learning, that one developer was not going to carry the other—a reasonable concern."

I've worked with other people whose reasons for a veto weren't as sensible and easily addressed. One developer on an interviewing team said, "I don't want you to hire anyone. I don't want anyone else working with me. I want to work alone." We found a different project for that person to work on, and I removed that person from the interview team. If the opposite happens and you have an interviewer who is consistently enthusiastic about every candidate, ask, "What was most exciting about the candidate to you?" This question is discussed thoroughly, along with others, in Chapter 10, "Following Up After the Interview."

Make sure everyone agrees on the work to be done and on the kind of person who can do it.

On the other hand, you, as the hiring manager, shouldn't make the decision if you can't obtain hiring consensus or if the situation isn't clear-cut. If the candidate is ranked "just okay" by the interview team, your best bet may be to interview more people, rather than hire someone about whom your group is lukewarm. And don't fall prey to pressure to hire. Threats such as "If you don't hire someone by March 15, you'll lose the requisition" are a trap. That's a sucker's game.

Don't train your management to pressure you into hiring someone before you're ready, and don't let an internal candidate's current manager make your hiring decision for you either. Make sure members of your team interview an internal candidate the same way they would interview an external candidate. Sometimes, your colleagues want to unload people and will be less than candid about an internal candidate's appropriateness for your group. Or, they may have the best intentions, but not know about your group's culture and whether the internal candidate will be a good fit for your team.

If you determine that you do not want to consult your interview team—for whatever reason—do not form such a team. That is, if you believe that you should make the hiring decision alone, ask yourself these questions:

1. *Is your competence on the job judged by your ability to make hiring decisions by yourself, without the input of others?* If so, discover who evaluates your job performance and your ability and see whether you can change his or her criteria.
2. *Will anyone else work with this candidate on a daily basis?* If so, why do you not care what your colleague thinks about the candidate?

If you could single-handedly learn everything you needed to know about a candidate, then it would be okay to make a hiring decision yourself. Since that is rarely possible, it makes sense to use the team to help make the hiring decision.

Plan what you will do if you can't find the right people.

Every profession has ups and downs with regard to hiring. During recessions, there may be many candidates from whom to choose. During boom times, the demand for people appears to outstrip the supply. That's when your hiring and management strategies are critical to your success.

You can choose one of several options when you can't find candidates to fill your positions:

- *Expand your search.* Make sure you're taking advantage of all the recruiting possibilities (see Chapter 4, "Sourcing Candidates").

- *Change your hiring strategy.* Hire people who have fewer specific technical skills, but who fit the culture and are fast learners or great problem-solvers—and then train them (see Chapter 15, "Moving Forward").
- *Choose which projects you're not going to do.* Alternatively, choose *when* you will do the projects.

Take a few minutes, and develop a hiring strategy. You'll find the rest of your hiring easier to accomplish. Following is a template to help you develop your hiring strategy:

Problem Categories & Problems to Solve	**No**	**Yes**	**Desired Characteristics & Problem Solutions**
We need more people to do more of the same kinds of projects.			Technical skills, as long as enough candidates exist. If not enough candidates, focus on people's ability to learn and teamwork.
We're making the transition from one kind of technology, work, or product to another.			Problem-solving skills, skills learning new technology, adaptability, and cultural fit.
We're on the cutting edge of technology.			Adaptability, cultural fit, and ability to work in teams.
We're putting together a brand-new group.			Experience working, experience applying functional skills to new product domain, experience creating a new team and making the team successful.
We're filling in with other skills to change what we currently do.			Cultural fit, fit with team, expertise in specific functional skills and ability to apply those skills to new product domain.
We want to make our projects finish faster.			Different functional skills, teamwork, and cultural fit.
We need a few people now, but not forever.			Consider contractors with great communications skills so you won't lose their work when they're gone.
We have to fire too many of the people we hire.			Verify that the interviewing team is composed of people who know how to interview and that they understand the requirements of the position. Use limited consensus to hire people.
Turnover is too high.			Review cultural-fit needs and verify that interview questions address cultural fit.
Recruiting more people is difficult.			Use multiple sourcing mechanisms. Make sure résumé-screening filter isn't too tight.
We need more diversity in our group.			Look for diversity in background, attitude, personality, product experience, as well as in race and gender. Look for different levels of experience.
We need more management capability.			Look for management skills along with cultural fit.

Table 1-2: Hiring Strategy Template.

Once you've listed your concerns, organize them in order of priority to help guide your job definition, recruiting, and hiring actions. Don't forget to explain your objective to anyone who helps you recruit or interview.

Review your checks in the Yes and No columns in the template above. Then choose which actions to take.

Points to Remember

- Know why you're hiring more people. Define your problems to define your hiring strategy.
- Know what types of roles you require. Do you need more developers, more support staff, or more testers? If you had more writers, could you work differently? Are there tradeoffs you can make to fill a specific role?
- Know how you will decide on which candidates to select for which jobs. Consider consensus-based hiring as the decision-making mechanism.
- Know that you need a risk-mitigation strategy. If you can't find the people you need when you need them, define what you're going to do.
- Re-evaluate your hiring strategy periodically, based on how much hiring you've completed.

2

Analyzing the Job

Hiring Manager: "I need to hire a developer/tester/writer."

HR Rep: "Okay. What does that person do?"

Hiring Manager: "Engineering . . . development . . . testing . . . writing, of course. A little bit of this, a little bit of that."

Most technical staff and technical managers do a little bit of this and a little bit of that, but unfortunately, that's not even close to a job description.

Performing a thorough job analysis is a fundamental component of the hiring process, and your analysis should help you define the requirements for the job. As a result of taking the time to analyze the job, you will discover criteria that you can use to create a job description. You won't need to embarrass yourself by only being able to tell a recruiter you're looking for "a little bit of this, a little bit of that."

You will want to perform a thorough job analysis if any of the following conditions apply:

- You've never analyzed your open requisitions before in this organization.
- You're starting up a group, after a reorganization.
- You need to change the general job descriptions of the people you're managing.

If you've defined your *hiring strategy* according to the approach detailed in Chapter 1, "Developing Your Hiring Strategy," you know the problems you want to solve. Now it's time to define which kinds of people you need at what level to solve those problems. You may already have a high-level idea of what you want to see in potential candidates, but performing the job analysis will enable you to see what specific tasks a particular employee will need to be able to perform. In the sections that follow, I discuss job analysis in the context of designers, testers, writers, support staff, and project members in general, but I do not address jobs in management. If you're hiring a technical manager, look at Chapter 14, "Hiring Technical Managers," for how to analyze a manager's job.

Let's start with the high-level tasks you probably already have in your mind. *Developers* may design, implement, peer-review, unit-test, and debug. *Testers* may evaluate designs and perform black-, white-, or gray-box testing. *Writers* may develop new material or they may edit someone else's written material. *Support staff* may answer calls and help customers by determining and fixing problems. It is also possible that the staff you will need may be required to perform different work altogether from the tasks I've listed. To enable you to identify what you want in a candidate, the job analysis must be more than just a description of the job's functions. Your job analysis should identify the reasons you're hiring someone for your particular culture and organization.

Before you start writing a job description, analyze the job in the context of your current team's skills, experience, and personalities, so you know what to look for in a candidate. I use the following steps to analyze the job:

1. *Define the job's requirements,* including defining the person's interactions and the type of work he or she will perform.
2. *Define both the essential and the desirable qualities, preferences, non-technical skills, and technical skills.*
3. *Define the required educational background and the desired level of technical experience.*
4. *Define the activities and deliverables,* the outcomes of the work you want the employee to perform.[1]
5. *Define the factors that could eliminate a candidate* from consideration.

Once I have filled in as much information as I can at each of the five steps, I then complete a job analysis worksheet (a sample is shown in Worksheet 2-1 presented a bit later in this chapter). The worksheet makes it easier for me to iterate on the analysis if necessary later on.

[1] For an excellent treatment of performance-based hiring, see Joseph Rosse and Robert Levin, *High-Impact Hiring: A Comprehensive Guide to Performance-Based Hiring* (San Francisco: Jossey-Bass, 1997), pp. 26ff.

Aside from large changes, numerous small changes can sometimes affect your hiring strategy and job analysis. Use smaller changes (such as a change in release cycles or overall volume of work) to trigger a reassessment of your hiring strategy and job analyses. I tend to use the yearly budget process and yearly performance evaluations to review job analyses, especially when I expect to be hiring people during the coming year.[2]

If you've never analyzed a job before, but have existing job descriptions, try filling out Worksheet 2-3 at the end of this chapter, and then match your job descriptions with what you've written in the worksheet. If they don't match, it's time to reanalyze the positions you have open.

One caveat: The perfect person for the job probably does not exist, or, at the least, cannot be hired for what you're willing to pay. That's okay; we all have to live within constraints in a real world. That's why analyzing the job is so critical. The analysis helps you decide which criteria are required and which are optional. You can then find a candidate whose qualities are close enough to the original criteria and choose what tradeoffs you will make in terms of position requirements, qualities, preferences, and skills.

Define the job's requirements.

Begin analyzing the job by creating the high-level view of the tasks to be done and define the kind of person best suited to do those tasks. Think about other people with whom this person will work, and consider the conditions under which they will work together—in effect, the customers or clients of the employee.[3] Identify the personal qualities and preferences of current staff members that enable them to get the work done properly—characteristics such as whether a person is talkative or quiet, or whether someone is innovative or prefers to follow strict guidelines. Define the specific tasks this newly hired person will do, the technical environment in which the person will work, and the kinds of deliverables the person will complete. If you also are looking for specific desirable experience—such as pharmaceutical company experience, recent college training, or the sub-

[2] Using your work analysis to define critical attributes can be difficult. If you are an experienced hiring manager, you may have heard of using KSAOs (Knowledge, Skills, Attributes, Other) as a technique for analyzing the job. I don't use them for two reasons: First, I haven't encountered them in any of the technical organizations in which I've helped hiring managers; and second, KSAOs are often underspecified so they are unhelpful to the hiring manager. For a discussion of these, see Rosse and Levin, op. cit., p. 34.

[3] Recognizing that a new hire will interact with users and customers can help you identify the best person to hire. For more on the topic, see Donald C. Gause and Gerald M. Weinberg, *Exploring Requirements: Quality Before Design* (New York: Dorset House Publishing, 1989), pp. 68ff.

contracting experience a military contractor might have—list that on the job analysis sheet. Or, if this person may need specific talents and skills to complete the work in your time frame, list that detail as well. When you spend time analyzing the job at the start of the hiring process, your recruiting and screening efforts are more effective.

List the job requirements to include five things: (1) interactions, (2) functional roles, (3) role level, (4) management component, and (5) activities and deliverables.

I find that if I start by defining with whom the person will work, before I list the candidate's deliverables, I'm more likely to develop a job description that I don't need to revise after I've seen the first few candidates, wasting precious phone-screen and interview time. No matter how you start an analysis, don't forget to include all the analysis components in your position definition.

This approach admittedly is a top–down approach, and may not be comfortable for you if you prefer to start with the details. If you prefer, try working from the inside out, starting with the role level, the role, and the interactions. Then, define the management component, activities, and deliverables. Don't start with the activities and deliverables first; you'll forget something crucial in the interactions, role, and level.

To define the requirements for candidates, use the following techniques:

First, define the expected interactions. Define with whom this person works. One way to think about the interactions is to consider who the person's customers are, and who the suppliers are. To whom does this person deliver work product or information? Define the daily, weekly, and monthly team and personal interactions. I tend to use titles of people, not names here, so that my job analysis is clear even if the group experiences turnover. However, if you do use actual names, the people named may be willing to help you review the analysis. Write down how the person in this position will work with other people: cooperating, influencing, providing work direction, negotiating, or working as part of a team. Specify the frequency of the interactions.

Second, define the functional roles for the position. Sometimes, you only need to record "developer" or "tester" here. But if you have a writer who also fields calls, the position is a combination of roles. I like to think of functional activities (architecture, design, programming, planning, analysis, quality assurance, release engineering, integration, documentation, and testing, for example) when I define the roles this position requires. Then, go on to define how large a part each role plays in the position.

If you're defining a job with multiple roles, spend enough time on this step to know which roles are most important, or how much time you want each role to take. People find it difficult to work when they have to spend time on two separate functions. If you're looking for a support representative who will also test, a writer who will also support, or a manager who will also perform development, decide what percentage of time you think is appropriate for each function, based on the activities and deliverables you need from the position. Keep in mind that once you've hired someone, the percentage may change.

Third, define the position's level. You may find it difficult to determine whether this is a senior- or junior-level position, especially if your organization does not have a formal, posted job ladder. I follow the guideline that the more experience and the broader the experience required, the more senior the position. When in doubt, look at the deliverables and activities, discussed below.

You might want the new person to change the balance of seniority in the group. If your group is top-heavy with senior people, you may want to start adding junior people who can be trained. You can make good use of a junior-level person even if the position is on a time-critical project that requires a lot of technical knowledge, simply by moving a current senior-level staff member to the new position and backfilling his or her current work with the new junior-level hire.

Fourth, define the position's management component. Many technical people take on technical leadership roles at some point in each project. However, some positions are combinations of technical work and technical leadership. Or, the position may have elements of project management or people management. Decide now if this is a purely technical job.

When you define a management job, define the scope of the job. For example, is the scope strategic (requiring a person with the ability to plan an organization's operations), operational (requiring someone with the ability to plan and oversee the daily work of the organization), or supervisory (requiring a person who has demonstrated administrative ability and who is task-oriented)? Then, define which functional area or areas are involved in the job. Analyzing a manager's job is more difficult than analyzing a purely technical job. See Chapter 14, "Hiring Technical Managers," for more information.

Fifth, define the position's activities and deliverables. Most people start with this fifth task when writing a job description, but I caution against jumping in without having done the previous tasks. Although it is true that a developer develops, a tester tests, and a writer writes, and that for some jobs, the activities and deliverables are that easy to define, most technical positions require more than just a one-word functional description of activities and deliverables.[4] The more specific you can make the deliverables, the better your job analysis will be. If you are hiring for specific work, be specific in the activities and deliverables, noting for example, "Complete the Big Project by Jan. 30," or "Lead the architecture effort for ModuleX, completing the initial architecture by June 5." The more specific your activities and deliverables are, the more people on your interview team can focus their questions.[5]

Developers may perform requirements elicitation, requirements analysis, model building, design, architecture, analysis of defects found, coding, debugging, unit-testing, and more—for a specific project.

Testers may perform test planning, test development, and defect report generation; they may gather and disseminate metrics reports about defects or performance, and more—for a specific project.

Writers may write documentation, edit, run tests, develop examples, develop tutorials, and more—for a specific project.

In addition to project work, each position has daily, weekly, monthly, and some yearly deliverables. When you consider deliverables, consider how you will measure the employee's success. If you think about what success means in this position, you will be able to define the activities and deliverables more easily. Be as specific as you can, using completion dates, module names, product or project names, or people's names. For example, by specifying "Mentor junior members of the tech. pub. group to design a new document for the BuyWrite project by December," you have defined both activities and deliverables for the open writer position. Write the activities and deliverables as if you were writing a yearly list of goals and objectives for an already hired employee.

Some position's deliverables are harder to define. If you need a technical coach or a facilitator, describe those services in terms of the benefit to the position's customer. One VP of engineering enjoyed benefits he attributed to his agile-programming coach's services: "He watched how people worked together. He looked for teams that weren't clicking. He made them click in a way that made the project manager happy." The VP also could have described the agile-programming coach's activities and deliverables this way: "Monitor the team—daily. Look for problems the team has

[4] See Rosse and Levin, op. cit., p. 27.

[5] An excellent treatment of the importance of identifying the critical performance objectives of every job appears in Lou Adler, *Hire with Your Head: Using Power Hiring to Build Great Companies*, 2nd ed. (New York: John Wiley & Sons, 2002), pp. 26ff.

using the methodology and other obstacles. Remove any obstacles to the team's success—daily." Here, the position is described in terms of benefits to the team members and project manager.

I find that as I define activities and deliverables, I look repeatedly at and review the other parts of the analysis, specifically the roles and the level. Defining the level of seniority needed for whoever fills the position may help you judge the activities and deliverables more easily.

Job Category	**Activities and Deliverables Needed to Satisfy Job Requirements**	
	Junior Level	**Senior Level**
Systems Engineer	Analyzes current requirements, looking for ambiguity.	Develops requirements cooperatively with others; assesses requirements documents for their completeness, fit with the rest of the system, and for ambiguity.
Developer	Designs modules after the major portion of the design is complete. Writes and compiles code in accord with some predetermined description. Arranges to have his or her own designs or code reviewed. Reviews peers' code. Uses configuration management system to check code.	Moderates code inspections, drafts design specifications, drafts functional specifications, and designs large pieces of the system.
Tester	Develops test procedures. Runs tests and reports on tests. Gathers metrics. Automates pieces of test procedures. Attends reviews and inspections.	Designs test approach. Designs automated test procedures. Moderates reviews and inspections. Develops metrics reports.
Writer	Writes documentation.	Plans tech. pub. and book design. Plans on-line help design.
Support Staff	Takes information for each incident. Resolves incidents that don't require looking at the code.	Resolves any incident. Manages customer expectations for fixes. Manages the fix process for urgent problems.
Release Engineer	Writes scripts. Creates builds.	Sets up the configuration management tool. Manages and anticipates storage needs.
Project Manager	Plans and organizes small projects. (Make sure you define what small means to you: four people and four months, fifteen people and eighteen months, or something else?) Performs basic risk identification.	Manages large-scale programs and systems, bringing project managers across the organization together to deliver an entire product, not just the technical part of the product. Negotiates with suppliers and customers.
Manager	Facilitates problem-solving. Performs strategic planning. Performs risk management.	Manages multiple groups. Coaches and mentors other managers.

Table 2-1: Junior- and Senior-Level Activities and Deliverables.

I make the distinction regarding junior- or senior-level requirements when I determine how much the position is worth to the company. The more senior the position, the more it's worth. For example, I'd expect to get a less-professional functional specification from a recent college graduate than I would expect from a senior architect. When I choose the level, I must be willing to pay for the work completed at that level.

If the company isn't able to pay what you think the position is worth, then be clear on the deliverables you can expect from possible candidates. If you expect too much of your candidates without offering enough pay, you may never hire anyone. Candidates will go through with the interview, conclude that you are asking too much for too little pay, and decline your offer.

Once you've described the deliverables, you're ready to look at the other necessary candidate characteristics.

Define the essential and desirable qualities, preferences, and non-technical skills for a successful fit.

The second part of your job analysis is to look at the qualities, preferences, and skills that will help a candidate succeed in the position and fit into your organization's culture.

Different companies can have different cultures. Take, for example, two organizations that are making high-performance commercial products. The first organization, SpeedyOne, is a typical entrepreneurial organization in which self-starting, responsibility-taking, and the ability to do three things at once are highly valued. Risk-takers are especially successful at SpeedyOne.

SpeedyTwo is an older organization whose emergence from start-up to established company was rocky, a factor that led to a significant aversion to risk in senior management. SpeedyTwo values employees with a passion for learning, the ability to make decisions by consensus, and the focus to finish projects. To an outsider, it would appear that both companies require similar technical skills. However, people at SpeedyOne prefer to work by taking initiative and forcing products to market quickly. The technical staff at SpeedyTwo uses consensus to make decisions, and uses technical practices such as peer reviews and walkthroughs to move their products to market. Each company looks successful and meets time-to-market needs, but the cultures differ dramatically.

There's nothing predominately right or wrong with how business is conducted at either company. The point to be made is that hiring managers must differentiate between the types of people who'll be successful in vastly different cultures.

Sometimes, the main reason a hiring manager doesn't hire a candidate is that he or she has a gut feeling that the person just won't fit well with the culture. But a "gut feeling" is not a good reason not to hire someone, so train yourself to articulate culture-fit differences. If you can't articulate why a person won't fit, you run the significant chance of hiring someone else with similar problems. Or worse, you hire people and then are vaguely dissatisfied with their performance, having expected "more at this level." If you have trouble articulating why candidates are not quite appropriate, then defining the qualities, preferences, and skills *prior to hiring* will help. If you're dissatisfied with some of your recent hires, take the time now to define the essential qualities, preferences, and skills that will fit your culture.[6]

Think also about diversity with respect to culture-fit issues. Sometimes, your group can be too homogeneous, so you want people who *don't* currently fit.

Essential qualities, preferences, and skills

A well-chosen employee who is successful in performing the job for which he or she was hired almost always will fit into the team and the larger organization. In addition, he or she will meet the technical challenge as well as fit the culture.

No matter what size a company is, culture can vary dramatically across the organization. You need to define the cultural needs as well as the technical needs for your immediate area, regardless of whether it sits within a corporation that is large or one that is small. Sometimes, a candidate who would be a great fit for one manager or in one organization would be a disaster for another simply because different managers and different organizations have their own unique styles. Review your group's culture to see which qualities are present in successful employees in your group, and compare how well candidates for the position match up.

Start your search for technical and cultural compatibility by assessing each candidate's *qualities,* such as initiative, flexibility, and technical leadership*:*

- **Initiative:** Do you need someone who looks for problems and fixes them? Do you want someone who is intellectually

[6] A thorough guide to defining culture-fit compatibility is Jim Harris and Joan Brannick, *Finding & Keeping Great Employees* (New York: AMA Publications, 1999).

curious? Do you need someone who can follow directions. Do you need someone you can train to do more and different work? *Define the amount and type of initiative you and your organization require from an employee.*

- **Flexibility:** Do you want someone who is completely flexible? Do you want someone who will develop rules of operation for you? Do you want more flexibility in a senior-level help-desk representative than you would require in a junior-level software developer? *Define the flexibility or adaptability required in the position.*
- **Technical leadership:** How much leadership do you want the new hire to take on? Do you want someone who is capable of creating new ideas, someone capable of sifting through ideas to discover the most appropriate solution, someone who will catalyze people to generate new ideas or decide on an idea, or someone who can follow through with the details so that an idea comes to fruition in a product? *Define the kinds of technical leadership you need from the candidate, keeping in mind that technical leadership does not necessarily correlate with years of experience and that not every employee needs to be a leader.*

After you have considered necessary qualities, consider a person's *preferences* for how he or she likes to work, and match them to your own preferences for how the job is to be done:

- **Procedural preferences:** Is it important that specific procedures be always followed to the letter on the job? Do you need people who take exceptional pride in following procedures? Can you tolerate mavericks who live to break the rules? *Define what procedural tolerance level you need.*
- **Tasking preferences:** Do you need a multitasker who likes to work on multiple tasks at one time, juggling, say, six tasks in various stages of "done-ness," or do you want someone who likes to handle only one at a time? Do you want someone who works to complete tasks, or a person who is happiest when required to context-switch between multiple projects or tasks? Do you want someone who can handle uncertainty, or a person who needs well-defined limits and schedules? *Define your tasking needs, but keep in mind that*

when looking for a multitasker, you want someone who can let you know when he or she can't take on additional work.

- **Goal-oriented preferences:** Do you want someone who can set and reach his or her own goals without much input from you or others? Can you deal with people who want to set their own goals rather than look to you to set the goals? If you prefer not to be a hands-on manager and choose to let goals evolve, can you manage someone who needs specific and detailed goals? *Define the goal orientation you require in an employee.*
- **Problem-solving preferences:** To what degree do you want people to own and solve their problems before they bring them to you? Do you want someone who will ask you to help establish task priority when confronted with conflicting tasks or schedules? *Decide how much independence is appropriate for the job and for your group.*
- **Learning preferences:** How much initiative do you want someone to take to stay current in his or her field? Does staying current matter for the particular job? *Determine whether the employee needs to have a yearning to stay up-to-date in all areas of technology or whether current skills are sufficient.*
- **Collaboration preferences:** Do you need a person who prefers to work alone or one who thrives when working with a group of people? Do you need a catalyst for a team? Are you looking for someone to complement the team? Does the position involve a significant amount of group work or very little work with teammates? *Define the amount of collaboration you require in the position.*

Finally, consider the *non-technical skills* that might make a person successful in your group:

- **Communications skills:** Do you need someone with excellent speaking or writing skills, or both? Do you want someone you can put in front of customers, a person who will be quick-thinking, good at fielding off-the-cuff questions, and the like? Do you require excellent phone skills for the job? *Determine your non-technical needs in the context of the specific job the person is being interviewed for—be cautioned, of course, about the legal ramifications of seeming to discriminate* *against someone who lacks specific non-technical skills when such skills are not required for the job.*

- **Performance-versatility skills:** Do you need someone who is strictly tactical and operational, or someone who can think strategically and plan what has to happen? Do you need a person who can handle projects of varying scope or a person who wants to focus on one kind of work? *Determine whether this person needs a variety of problem-solving skills, or a narrow range of skills focused in one product area or one kind of project.*
- **Negotiation skills:** Does the person need to be able to work with people inside and outside the group (or the company)? Is there management by authority, by influence, or both? Does this person need to manage choices between competing designs? Do you need someone who can negotiate with potential customers or different groups within the company when defining requirements? *Determine what level of skill your new hire will need as a negotiator.*
- **Problem-solving skills:** Will the new hire need to think about problems in a variety of ways? Can you use someone who takes the first solution that presents itself? How much creativity do you need in this role? *Determine what problem-solving skills are needed for the position.*

Not all the cultural qualities, preferences, and skills mentioned here will have the same degree of importance to you as they do to someone else. You may decide to select other qualities, preferences, and skills that your technical staff needs in order to succeed. Think about the successful people currently working in your organization and identify the qualities, preferences, and skills you find most valuable.

Each corporate culture is different, so define your essential qualities, preferences, and non-technical skills for your open position. You may realize that you have a particular type of person you choose to work with.

Following is a sample worksheet that can be used as you define the qualities, preferences, and skills required in a particular job. Add other characteristics required or desired for the job, using the Notes column to describe how those qualities, preferences, and skills fit the requirements of the job.[6]

[6] For talents that you may want to include when you define qualities, preferences, and non-technical skills, see Marcus Buckingham and Curt Coffman, *First, Break All The Rules: What the World's Greatest Managers Do Differently* (New York: Simon & Schuster, 1999), pp. 251ff.

Quality, Preference, or Skill	**Required**	**Desirable**	**Notes** *(Cite any required quality, preference, or skill specific to the job.)*
Quality: Initiative			
Quality: Flexibility			*This project manager needs to manage projects that require different lifecycles.*
Quality: Technical leadership			
Quality: Responsibility and independence			
Preference: Ability to work on multiple projects at one time			*The tester needs to be able to juggle planning for one project while developing tests for another project.*
Preference: Goal orientation			
Preference: Passion for learning			*We need someone who wants to keep up with the literature as new things are happening in the field all the time.*
Preference: Teamwork			
Skill: Communications skills			
Skill: Ability to handle projects of varying scope			
Skill: Influence and negotiation skills			
Skill: Problem-solving skills			
Add your qualities, preferences, and skills here.			**Add your notes here.**

Worksheet 2-1: Matching Qualities, Preferences, and Skills with Job Openings.

Desirable qualities, preferences, and skills

For many open positions, some of these qualities, preferences, and skills are not essential, just desirable. As you analyze the job, decide whether any of these factors would make you reconsider a candidate's pay, level of experience, or cultural fit in terms of the position for which you're hiring.

For example, technical leadership might be an essential quality in candidates you're recruiting for a position. The ability to help you develop and present project or program status to management would be desirable, and you'd be willing to pay at the high end of your salary range to get it.

Identify corporate cultural-fit factors.

Once you've fully described the qualities, preferences, and skills you want in a candidate, think about cultural fit. Your company has characteristics specific to it that will make or break a candidate's cultural fit. Not every organization is perfect for every person—and vice versa.

Think about your company's factors as you continue preparing your job analysis worksheet. People choose to work for companies for a variety of reasons, some of which have little to do with the technical work of the specific job.

Possible company-fit factors appear in Table 2-2, below.

Company Fit-Factors	**Preference Possibilities**
Working environment	Some candidates like offices with doors so they can have discussions with others without disturbing their colleagues. Some candidates prefer cubicles or a bullpen office, where people are accessible.
Career growth paths	Candidates may care about upward mobility in your company after they learn about your products and produce good results for you. If your company has not yet developed a technical or managerial job ladder, suggest that your Human Resources or Personnel staff do so now. Be ready to explain to a candidate your company's growth paths. Some people like to work where they can acquire more skills, prove themselves valuable, and then change roles to the next phase of their career at the same company. If this is beyond your scope or influence, ask your boss what to say to candidates.
Start-up or established	Some candidates only work for start-ups. Some only work for established companies.
Your products	Some people won't work for a company that produces tobacco products or munitions or parts that end up in weapons. Some people don't care about the product as long as it has a GUI or a scheduler or a compiler, for example. Some people only care about working in a specific industry, or on cutting-edge technical products, on consumer products, on end-user products, or on technical-user products.
Industry leadership	Some candidates like to work for industry leaders, or companies they think will be industry leaders soon.
Training policy	Some candidates look for reimbursement for education or other training opportunities.
Competing local employers	If you are one of only a few employers in a small town, it may be that none of these factors weigh as heavily for a candidate as the possibility of steady employment.
Company's profitability	Some people like to work for companies that are floundering because they enjoy the challenge of turning a company around. Some people prefer to work for a company that has a healthy cash flow and provides more security.
Company growth	Many candidates look for positive growth because there is the assumption their jobs will grow and they will continue to be paid. If the company's growth is stalled or negative, you can attract candidates who pride themselves on their ability to come into a negative situation and make it positive. Some candidates will be oblivious.
Cash flow	If the company is wealthy, candidates may assume that the company will buy necessary tools and provide training.
Recent layoffs	Some candidates will not interview at a company that's recently laid off large numbers of employees. Some candidates like the idea of a fresh start.
Recent acquisition or merger	Some candidates see acquisitions and mergers as a time of reduced productivity and increased drudgery. Others see them as a time of opportunity.
Overall corporate culture	Some cultures are dynamic and appear chaotic. Other cultures are calmer and more relaxed. Some candidates choose to work for a company because the projects and the organization are everchanging. Others choose to work for a competing company because the products and organizational structure is stable.

Table 2-2: Company-Fit Factors.

It's common among technical testers and support staff to want to move into development. I've hired people into testing or support, trained them for six months, and then let them move on. Other hiring managers want a guarantee that the employee will be in their group for twelve or even eighteen months. As a seasoned hiring manager, I'm willing to live with having an employee in my group for fewer months because I've had the chance to train the employee and I can use him or her to help me interview for a replacement. Whatever your position is, make sure you can articulate it, and that your position matches your company's policy on job changes.

One company's employment policy may be attractive to some candidates and repulsive to others. I once worked for a company that would not permit smoking anywhere on its premises. I found this attractive, but the smokers I phone-screened did not appreciate the policy.

Some companies expect their employees to participate in a variety of extracurricular activities, such as raising money for charitable organizations, weekend sales meetings, social events, and so on. Candidates who find these activities exciting, and view them as a positive use of their time, are an appropriate fit for those companies. Candidates who do not want to participate in these extra-corporate activities are not a good fit. Avoid wasting your time interviewing people who aren't a good cultural fit, especially if the lack of fit is a misalignment of cultural values regarding where employees should spend their time.

If a candidate's success hinges on these factors, then remember to ask questions about these specific cultural factors during the phone-screen and also during the interview. See Chapter 7, "Developing Interview Questions and Techniques," for suggestions.

Define the corporate cultural-fit factors so that you can use them in a job advertisement. You'll have more success attracting the most suitable candidates if you identify such factors right at the start. Even if you don't want to use the factors in an ad, being aware of the cultural-fit factors will help you answer candidates' questions about the company.

Define the necessary technical-skill level and the required educational background.

The third part of the analysis involves defining the required minimum educational background a candidate needs to be successful in your organization, and the minimum technical skills required for the candidate to complete the deliverables. Consider the work you need performed and the background required to perform it. Determine if your organization has assumptions about required educational background for technical people.

Prior work experience

A candidate with *prior work experience* can be very valuable, but you'll want to look beyond what is stated on the résumé. Define the work experience required of a candidate before you consider him or her for the job. Consider functional skills, product domain, tools and technology, and industry experience.

Functional skills experience

A candidate who has *functional skills experience* possesses a technical understanding of how to perform the work required in the position. When you define functional skills experience, define what evidence you'll require to confirm that the skills you're looking for are present.

For developers, functional skills include proficiency in reading and writing code, as well as skills in designing, programming, and debugging. For testers, functional skills include boundary-condition testing and equivalence partitioning. For writers, functional skills include proficiency in grammar. For project managers, functional skills include knowing how to apply lifecycles to a project, as well as how to estimate, measure key project metrics, and run a meeting.

Functional skills experience is both technology-independent and tools-independent. Functional skills experience is gained in school and on the job. However, candidates may need specific functional skills to be successful in *your* environment, such as understanding how database schema are developed and used. Or, you may require a person with varied knowledge of data structures, or how to create and use watchdog timers, or with an understanding of how to avoid race conditions in real-time systems. If the position requires specific functional skills, include those skills in the activities and deliverables, for example, noting that you need someone to "design and implement the watchdog timer for ModuleA by June 1."

Product-domain experience

For a candidate to have *product-domain experience,* he or she must have knowledge of the product—either of the problems the customers want solved by this product, or the internal architecture of the product, or both. You may want someone who already understands how the product works and why your customers use the product. Or, you may want someone new to this kind of product, so that you receive the benefit of fresh eyes and attitude. For example, when I worked in machine-vision companies (companies whose products use cameras and software to "see"), our goal was to hire developers with machine-vision experience, so that they already understood the domain issues. When we wanted to hire testers, we needed people who had real-time and embedded systems experience, but not necessarily machine-vision experience, because we wanted the testers to be open to alternative testing possibilities.

When you assess your hiring needs, judge each candidate's product-domain experience in the context of your products. When you define a position and assess candidates, judge their experience against the products they've already worked on.

I identify product-domain experience as either problem-space or solution-space:

- With *problem-space domain expertise,* the candidate understands how the users use the product. Candidates generally have exposure to and a general understanding of product externals. They understand how the product interacts with the users or, from a black-box level, how the products' interfaces work to solve the customer's problem. Technical people acquire this level of expertise quickly.
- With *solution-space domain expertise,* the candidate understands how the product works on the inside—that is, the architecture of the product. We expect developers to have this expertise; if they don't, we expect they can acquire it by reading the code. Writers can acquire this expertise, depending on the kind of documentation they write. Testers and technical support staff members can acquire this kind of expertise if they can read how the product works on the inside, or if they are taught about the internal workings.

A candidate who has the ability to acquire both problem-space and solution-space expertise is likely to be most valuable to you. As you define the job (and later, as you review résumés or interview candidates), determine how much expertise the candidate must have both in how the product works (solution-space expertise) and in how to solve the problems of the customer (problem-space) using his or her functional skills. People with both kinds of domain experience understand the specifics of how the product works, and can understand how to relate their functional skills to improve the product. At the senior level, developers, testers, designers, and architects all are likely to have this expertise. It's easier to obtain this level of expertise if the candidate already can read and write code, but it's not required. All technical staff members can acquire both problem-space and solution-space domain experience if they have the ability to understand the product and are educated about it.

How the different kinds of domain expertise work together is illustrated in the following example: If you have a multi-threaded product, one in which numerous instances of the product can be run simultaneously on one machine for multiple users, then a candidate has reached problem-space expertise if he or she knows that the product is multi-threaded and why the product requires multi-threading (for performance or throughput or whatever). With solution-space expertise, the candidate knows what kicks off multiple threads, and understands how to use that knowledge to develop pieces of the product appropriately or to create tests that test the product differently than if the product were single-threaded.

Technology and tools experience

Having experience with the hiring company's specific programming language, operating system, database, or other tools gives a candidate somewhat of an advantage over applicants without suitable *technology and tools experience.* Mastery of the specific technology and tools can be easily taught to a candidate who is fully qualified for the job in all other areas. However, technology and tools experience may or may not be a consideration in whether you or your company will decide to hire a candidate. The decision more likely will depend on how quickly you want him or her to start being productive in your environment. For example, if you're starting up a company that will use an Oracle database for a server and a Java front-end for a client, you want to hire people who know how to use Oracle databases and Java front-ends, as well as how to use the specific architecture to design, develop, test, document, and support that kind of a product.

If you're not doing the phone-screens and interviewing yourself, make sure you specify to the people who are performing those tasks whether you need to hire a person with experience using a specific tool or someone with general tool experience. For example, if you're looking for a person to perform test automation, experience with any of the test automation tools may be acceptable. However, if you're looking for a person to coach your team members as they write automation using a specific tool, then experience in that tool is probably necessary. Similarly, a project manager with experience using any of the project scheduling tools may be acceptable, unless you also want that project manager to teach new project managers how to use the tool. Teaching, coaching, and mentoring activities may require a good working knowledge of a specific tool. Otherwise, general tool experience is probably sufficient.

Industry experience

A candidate who has *industry experience* understands not just who your customers are, but how your customers will react, and what they expect from their systems. Industry experience relates to how people use the product; product-domain expertise is knowledge of the products' internals.

You may want to hire a candidate with experience specific to your industry so that he or she understands your customers and their expectations, and has an understanding of the types of problems you encounter in your work. For example, people who work on software for airplanes understand that their products will be audited and their processes will be assessed to ensure that the product development group hasn't created an unsafe environment. Someone who's worked in the shrink-wrap, commercial-product-development world might not welcome all these assessments and audits, but a veteran of the aeronautical industry must be more accepting of them. The pharmaceutical industry is another example of an industry in which process rigor and audits are common practice.

Focus on the experience necessary for the job you have to fill. Don't worry about planning too far ahead—it's too hard to predict the future. I've found that when I plan too far into the future, my candidates don't have the skills to perform the work that I need done now.

Identify essential technical skills.

Only you know whether it is important to fill the current open position with a candidate who will know how to use your technology immediately. We all spend time training people to be successful in our organizations, but are you planning specific skill-based training in addition to helping the person navigate the unfamiliar seas of your workplace?

When you define the technical skills required, make sure you know what's actually required for the particular job. For example, if the product is written in Visual C++, you may require someone with a number of years of Visual C++ experience. The number of years of experience you require should depend on whether this is a senior- or junior-level role. If a working knowledge rather than in-depth experience is required, then you may not need to specify a minimum number of years of experience in the job analysis worksheet and the job description. However, if you are seeking a mature candidate who has worked on numerous products, then specify overall years of experience. If you're seeking a person with in-depth technical knowledge, then look for someone with a few years of specific experience instead of a variety of technical skills spread over the years. One way to define the required technical knowledge is to ask your current team members to help you define the requirements.

When specifying technology, remember to consider your specific development environments. Someone who has developed software in the C language using a UNIX operating system may have a different idea of how to develop software than someone who has developed software using Visual C++. In the job analysis, the job description, and also in any advertisements, specify the minimum number of years of experience you want an applicant to have in each environment.

When you consider functional skills or product-domain experience, think about whether you want someone with experience throughout the entire project lifecycle. Someone who's lived through a product release will have had a different experience from someone who's worked only on canceled projects or who has been moved off projects before their completion.

Carefully consider which skills you need a candidate to possess, and which skills you are willing to provide by training the person after he or she has become an employee. In a highly competitive job market, it may make sense to hire candidates who have appropriate problem-solving skills, who are adaptable, and who demonstrate an ability to learn, but who don't necessarily have experience in the specific operating system or programming language they'll need to use. In most cases, you will be able to train them in the required programming language in less time than it could take to wait for just the right candidate to cross the threshold. In a less-competitive job market, if you have specifically described the necessary technical skills in the job analysis, job description, and all advertisements, you will reduce your recruiting time because you will narrow your field to appropriate candidates only.

Avoid the appearance of requiring applicants who have more experience with a specific tool or technology than you can reasonably expect or than you truly need. When object-oriented programming came into vogue, reliable, commercial compilers had only been on the market for about a year, but some companies were requiring job applicants to have a minimum of five years of C++ experience. This kind of unreasonable requirement only encourages candidates and external recruiters to stretch the truth, or equally problematic, to not send you their résumés. You, and anyone else involved in describing an open position, need to learn enough about how your company uses technology to hire people for your group.

It's easy to use years of experience as a shorthand indicator of the experience or knowledge a candidate would need to be successful in a position. Nevertheless, when you analyze the position's needs in terms of work experience and technical skills, you must ask yourself what you mean by "work experience."

When you've defined your technical experience requirements, add them to the bottom-most boxes in the job analysis form, as shown in the worksheet portion replicated from Worksheet 2-1:

Quality, Preference, or Skill	**Required**	**Desirable**	**Notes** *(Cite any required quality, preference, or skill specific to the job.)*
Skills: Technical. One year of ClearCase administrative experience.	Required		*We use ClearCase and do not have the budget to train a novice admin. for this position.*

Worksheet 2-1 (continued):
Matching Qualities, Preferences, and Skills with Job Openings.

Technical skills confusion?

You may sometimes need to fill a position that is common at other companies, but that does not already exist at your company. Or, perhaps you know some of what you want done in a job, but not enough of the requirements to feel comfortable about listing the job opening. Perhaps you need to hire a technical support manager, and you know something about technical support, but you do not know enough to describe the essential skills required. If you are faced with such a problem, try the following approach:

- *Ask for help from someone who might know what the job entails.* Such a person might be someone who either has done the work or has successfully hired people for the job—other managers at your company, or outside colleagues, for example. If you're using internal or external recruiters, ask them for help. Ask for advice from consultants, an academic advisor, your mentor, or from someone whose Web page interests you and is relevant to your industry or technology. The more people you ask for information, the better able you'll be to refine the job analysis and the closer you will get to determining the essential job functions.
- *Use analogy.* If you don't know what job a particular job title describes, approach the job analysis from the point of view of the tasks that need to be done. For example, software companies began using configuration management systems in the 1980s. In the early 1990s, the technology was still new

to the software community, and there were not enough release engineers and configuration managers to fill the open jobs. However, people did know what tasks needed to be done to create bills-of-material for software products, for example. So, while these managers were not called release engineers or build engineers or configuration management engineers, they knew what they needed to perform and were able to define the essential job functions.

- *Ask your Human Resources or Personnel Department staff members to compare the offered salary and compensation package with that printed in industry-wide surveys.* Companies that participate in salary surveys have access to lists of job titles and job functions. You may be able to use these lists as a place to start defining the essential technical skills.

Identify desirable technical skills.

While analyzing the job, you've probably thought, "I'd like this-and-that technical skill, but I don't require it." Now you can add those desirable skills to the job analysis.

A True Story

Donna, a Tech. Publications manager, was defining a writing position. She had already determined that a desirable skill was "project-management experience." The writer was going to work on a project in which the project manager was stretched too thin, and none of the other groups had people who could help with management of the project. Donna was ready to pay more in salary if she found a writer who could also help the project manager manage the project. Donna listed "project scheduling and coordination" as desirable skills.

The moral: When defining desirable skills, don't forget to consider your tradeoffs. You may want to change the job level and salary depending on whether you find a candidate with more or fewer desirable skills.

Evaluate educational or training requirements.

Doesn't every technical person need a college degree, at the least? No. A college degree shows a kind of perseverance, but not necessarily the kind of perseverance you need in the person you hire to perform your work. College degrees awarded in the fields of science, engineering, or computer science indicate that the candidate may have learned enough technical information to understand the job to be performed, but degrees don't mean the candidate can perform as needed. Don't let the presence of a degree convince you that a person has the skills and characteristics to do the job well. And don't let the lack of a degree deter you from screening and interviewing people whose experience looks like it might fit your opening.

A candidate's experience with successful product development, release, and support can be more valuable to you than a college degree. One of the best test developers and all-around system administrators I ever worked with was someone who began programming at the age of eighteen and didn't bother going to college until he'd reached his late twenties. Some of the best developers I've worked with never graduated from college. Many good managers I know never even finished college, let alone obtained an MBA or any other advanced degree.

If your HR Department has a policy against technical candidates without degrees, talk to anyone who will listen to discover whether degrees are shorthand for describing some level of competence or experience. (In my view, the real key is whether the person has learned anything from whatever education or experience he or she has had.) Then, decide whether to either fight the policy or live with the restriction.

Experience catches up with formal education. I have observed that technical-degree holders lose technical proficiency if they don't use the particular skill or don't keep up with advances and changes in the field. Remember, too, that not all schools teach the latest technologies, practices, and techniques in their undergraduate curricula.

Depending on your culture, academic credentials may be essential. If you're hiring for a research environment, and your internal customers measure your staff members by the degrees they hold, look for people with degrees. If you're creating a professional services organization, and external customers will want to know how many and what kinds of degrees staff members have, your staff members may need a surfeit of advanced degrees.

A candidate's education or training only tells you the candidate has gone to school. What you want is to find the candidate who has learned to think. Education or training doesn't tell you whether the candidate has learned anything applicable to your job.

Licenses and certifications

What should you know about licenses and certifications? The first fact to remember is that governing bodies—usually the state or city in which the skill is practiced—oversee professional licenses, obliging the licensee to assume legal responsibility for work performed under the license. For example, a licensed "Engineer" has a legal responsibility for the quality of any design he or she signs off on. If you require someone with a license, then specify that license in the job analysis.

Professional organizations, rather than governments, oversee certification, but certification confers no legal guarantee for the quality of products produced by certificate holders. Certification is an indication that the candidate has experience in the field, was motivated enough to pursue the certification, and has mastered enough material to pass an exam. Unfortunately, even though certification may require work experience, the bodies that grant certification don't verify that the work done by the certification holder was successful or is even applicable to your needs. Certification does indicate that the candidate has learned the material well enough to pass an exam, but it carries no guarantee that the candidate can apply the knowledge to his or her work.

I personally do not consider certification to mean anything much when I am hiring someone for a technical position. Because the knowledge tested is functional-skills book knowledge, make sure you understand what the person must do to maintain his or her certification and the value of that certification to your environment. Also, determine if you will need to make any accommodations for the employee to maintain the certification. Many certifications require ongoing education in some form, so you need to be clear who will pay for that.

List the license or certification requirement as part of your job analysis if you do require either or both.

Define all elimination factors.

Now that you've described the job requirements, consider other requirements that would *eliminate* candidates from successfully performing the open job.

Each position may have factors that eliminate otherwise suitable candidates, that is, candidates who otherwise have the necessary technical skills and fit the culture. If you include these factors in your job analysis, you can build them into your job description, and avoid interviewing someone who seems perfect but can't meet some of the non-technical requirements of the job. Frequently, elimination factors fall into the job and company categories shown in Worksheet 2-2, below:

Possible Elimination Factor	**Elimination Category**	**Issue**
Travel	Job	Do you need someone who can travel half the time or even part of the time?
Availability	Job	Do you need someone who can commit to core hours or off-hours? Do you need someone who can rotate time on the telephone (common to technical support jobs), such as evenings or weekends, not just during the day? Do you need someone who is at work by a fixed time each day?
Relocation	Company	Are you willing to pay for the candidate's relocation? If you've posted the requisition on the Internet, people outside your geographical area may send you résumés. Some people will pay for their own relocation, so don't automatically reject candidates outside your immediate geographic location, but decide in advance whether you will pay for relocation.
Personal Characteristics	Company	Are there people your company chooses to legally discriminate against? As long as who you don't want to hire is not part of a protected class, such as people older than forty or a member of a minority group, then you can make this characteristic an elimination factor. If your company has a specific reality to consider, then you can use this reality to eliminate potential candidates who do not fit.
	Job	For technical support people who will handle telephone help-desk calls, the ability to speak clearly and audibly is frequently a plus. Decide whether you are willing to train people to speak more clearly, or whether you want to avoid interviewing them altogether.
Salary Requirements	Job	Salary requirements, additional benefits, and other perquisites should be defined in the job analysis and discussed with the candidate during the phone-screen so that such components of the job offer do not become a problem.

Worksheet 2-2: Job and Company Candidate-Elimination Categories.

Define the elimination factors for your job on the job analysis worksheet, and use the elimination factors in your phone-screen, sometimes as the first questions you ask.

Think twice about elimination factors.

Make sure when you consider elimination factors that you are not eliminating people who are different from you *simply because they are different*.

Diversity in an organization takes many forms: Product experience, gender, culture, and race are only a few of the areas where people differ from one another.

In large companies especially, elimination factors may create an unintended but real discriminatory hiring practice. Be sure to ask your corporate lawyer or someone in your Personnel Department or on the Human Resources staff whether your elimination factors might hinder diversity in your workplace. As an example of a potentially discriminatory hiring practice, consider the following: You work for a large multi-site organization. You want to hire a manager to oversee four geographically dispersed sites. You ask the question: "Are you available to travel half of the time?" That's the correct question, but you may well be discriminating against a candidate who could successfully manage the job without traveling at all. Consider whether you can specify the position without the travel requirements to allow people who won't or can't travel to apply for and appropriately fill the position.

Travel can be a problem for non-managers as well. If your organization supplies on-site support to customers in a variety of locations, you may believe you require a customer-support engineer who can travel half of the time. However, by specifying travel as a requirement, you may be ruling out primary-care-givers, physically handicapped people, and more. That can be illegal. Instead of requiring travel, consider setting up alternatives such as videoconferencing, local support staff, self-diagnosing hardware, and so on.

Frequent travel may be a requirement for various categories of technical staff—systems architects, project managers, systems engineers, product managers, and senior-level designers and managers, for example—but make sure to note the reason. If you need architects or designers to travel for a week each quarter to develop the next-generation product line with their peers, that's a different travel requirement than requiring a systems engineer to travel three weeks out of every four to a customer to elicit requirements. If you note why you require travel, you can develop effective interview questions. Then, if a candidate asks for the reasons behind the travel requirement, you can quickly and easily explain the reasons.

Make sure your elimination categories do not exclude handicapped people from your hiring process because of their disability. Discriminating against disabled people in the United States (and most other countries) is illegal, of course, but it is also foolish and shortsighted: A physical handicap most probably will not impede mental performance, and a mental handicap most probably will not impede performance of a primarily physical job.

Now that you've analyzed the job, it is time for you to complete the job analysis worksheet so that you can create a precise and practical job description.

Complete the job analysis worksheet.

Use your worksheet as a place to capture the job analysis work you've done so far. Describe job components, cultural attributes, education, and company factors, and use what you have recorded as a basis for writing a job description.

Once you have filled out the worksheet provided as a sample below, take a look at the case study example following it. The Walker Software Case Study appears in bits and pieces throughout the book and in Appendices A and B.

Defining Questions	**Needs & Observations**
Who interacts with this person? What roles does this person have in this job? What level is the company willing to pay for? What's the management component?	
What are the job's activities and deliverables? What periodic deliverables are required?	
What are the *essential* qualities, preferences, and non-technical skills? Initiative? Flexibility? Communications skills? Ability to handle projects of varying scope? Ability to work on multiple projects at one time? Influence and negotiation skills? Goal-orientation? Technical leadership and problem-solving skills? Responsibility and independence? Passion for learning? Teamwork skills? Others?	
What are the *desirable* qualities, preferences, and non-technical skills?	
What are the *essential* technical skills? Technology/tool skills? Functional skills? Product-domain skills? Industry experience? Others?	
What are the *desirable* technical skills?	
What minimum level of education, training, or experience is required?	
What are the corporate cultural-fit factors? What benefits should be offered? Company growth? Cash position? Industry leadership? Entrepreneurial environment? Benefits? Company size? Others?	
What elimination factors should be considered? Travel? Availability? Salary? Others?	

Worksheet 2-3: Job Analysis Worksheet (with Job Requisition Name).

Case Study: Walker Software

Now let's walk through the job analysis worksheet using an example from Walker Software, a fictional hundred-person company that manufactures an add-on to the telephone company's central office switch. Walker is coming out of its start-up phase and is becoming profitable, but members of the management team are cautious about growing too rapidly—they have only approved hiring people they can support with the current revenue stream.

Who should we know about on Walker's current staff? First off, Vijay is Director of Engineering and Operations and is responsible for product development and support, with a total staff of thirty people. His direct reports are Dirk, a development manager; Susan, a test manager; Ed, the support manager; and two project managers. Vijay's problem is a simple and common one—he needs more people to do more of the same kind of work.

Dirk manages eighteen people in development, five of whom are senior-level developers and technical leads, eight of whom are mid-level developers, four are junior-level (but not entry-level) developers, and one is a release engineer. Dirk is looking for a junior- to mid-level developer.

In the test group, Susan manages four people now: One is a technical lead/senior-level tester, two are mid-level testers who can read and write code, and one is a junior-level tester. She's looking for one mid-level test automation engineer and a mid-level black-box tester.

Ed has five people in support: Two are tier-1 engineers, two are tier-2 engineers, and one is a tier-3 engineer. Ed is looking for an additional tier-2 engineer.

Also on Vijay's staff are two project managers, but he wants to start up a third project and needs another project manager. The technical people are matrixed into the projects (they report to a functional manager, but perform day-to-day work for a project manager), so the project managers need to be relatively senior, to deal with the issues involved in selecting and negotiating for the appropriate people for their projects.

Dirk's first-draft Software Developer job analysis worksheet is shown below.

Defining Questions	**Answers re: Needs & Observations**
Who interacts with this person? What roles does this person have in this job? What level can we pay for? What's the management component?	*Works with developers, testers, writers, and project manager as a software developer. Mid-level, no management component.*
What are the job's activities and deliverables?	*High-level design (post architecture), implementation, participate in requirements-definition meetings, moderate and attend design and code reviews, contribute to smoke-test suite, develop integration tests. Able to manage subsystem development under the guidance of a technical lead.*
What are the essential qualities, preferences, and non-technical skills?	*High collaboration, high teamwork, adaptable, able to consider multiple designs.*
What are the desirable qualities, preferences, and skills?	*High focus.*
Essential technical skills	*C++, UNIX system calls, UNIX shell scripts. Data structures. Understanding of telephony industry.*
Desirable technical skills	
What minimum level of educational or training requirement is needed?	*Four years working as a developer, at least two completed projects. BS nice, not necessary. BS equivalent okay.*
What are the corporate cultural-fit factors?	*Small company, project-oriented, high growth expected, stock options.*
What elimination factors should we consider?	*$70,000 cap for salary.*

Worksheet 2-3: Dirk's Job Analysis Worksheet for the Software Developer Opening.

Dirk, Susan, Ed, and Vijay are all looking for someone with at least four years of technical experience. Actually, they're each looking for someone who's completed a variety of smaller projects, and putting the experience level at four years is their shorthand way of saying so. Vijay is looking for someone who's already managed at least two projects from start to end. The qualities, preferences, and skills are similar across group lines, but they're not the same; for all of the job analyses, see Appendix A.

Points to Remember

- Develop a job-analysis method for defining the position's requirements. The interview team will use the requirements to evaluate a candidate's work and cultural fit.
- Consider the context of your group and company when you analyze the job.
- A person's qualities, preferences, and non-technical skills have a huge impact on his or her ability to work successfully in your organization. Understand what's important to the company and to your group.
- Don't make certification or formal education the basis for a job description or for a hiring decision unless your culture requires the degree or you're building a professional services organization.
- When you require the services of a licensed professional, make the license a required component listed in the job analysis.
- Remember your company's attributes when analyzing the job. What appeals to one candidate may not appeal to another, and you want to make sure you attract the candidates to whom your company will best appeal.
- Use elimination factors to eliminate people who won't work out, even if they have all the other technical skills, qualities, and preferences to succeed in your culture. You will be doing yourself *and* the candidate a favor if you address these factors before the candidate accepts the job. Do not use elimination factors to discriminate.
- Record your requirements on a worksheet, so you can refer back to them when you want to hire again.

3

Writing a Job Description

> *"Hey, Jack, did you see the job description SuperSoft posted? You'd have to have worked twenty years just to obtain the technical skills they want, and they only want a junior person. What planet are they on?"*
>
> —Disillusioned job-hunter

The job description for your open position has a specific purpose: It helps you identify candidates with appropriate qualities, preferences, and skills, and it enables you to screen out unsuitable candidates. A well-written job description helps you screen candidates by identifying the technical and non-technical aspects of the job.

If you don't perform the analysis first, you run the risk of missing essential cultural qualities, preferences, skills, or deliverables. Separating the analysis from the description is the same as separating the requirements from design in a project: You develop a different perspective on the problem, and you're freer to iterate on both the analysis and description. Remember that for every two minutes you spend iterating on the job analysis and job description, you probably save yourself at least thirty minutes by not having to phone-screen unqualified candidates; you may even save as much as sixty minutes by not conducting an interview you shouldn't have scheduled.

With a completed job analysis, you have the building blocks for a job description.

The job description is the way the hiring manager communicates the job requirements to the interview team and any recruiters. If you choose to do so, you may also use the job description to communicate job requirements to the candidates.

In addition, you also can use the job description to generate your internal job postings and external advertisements and to guide your interviewers when they develop and ask their interview questions.

Write a clear job description.

In Chapter 2, "Analyzing the Job," you developed a job analysis worksheet to define the job requirements. Now, use those requirements to write a job description. Use a template for the job description such as the one in the sample shown below. (Note that the one in the sample assumes that your audience is either internal or outsiders familiar with your company's culture and hiring tendencies, such as external recruiters.) Categories named in the template are explained in the paragraphs that follow it.

Job Description Template

Job title:

Reporting-to manager's title:

Generic requirements:

Specific requirements:

Responsibilities:

Elimination factors:

Other factors:

Job title: Identify the job by title, being as precise as possible, especially if departmental peculiarities must be taken into consideration when filling the position.

Reporting-to manager's title: Specify to whom the job holder will report, by name and by title. Being specific to this degree will prevent people from confusing your job opening with someone else's.

Generic requirements: List the minimum level of education or training required for this job level at this company. Do not bother describing levels that are industry-specific but are not relevant to the opening you wish to fill. By knowing the minimum generic requirements, you and your recruiters can more easily screen out unqualified candidates. List requirements you would expect anyone in the field to have, but be sure they are compatible with the essential technical skills identified in your job analysis. An example of a generic requirement might be stated as "two years of experience in a mainstream software language such as C++."

Specific requirements: Based on your job analysis, which indicated who would interact with this person and the essential technical skills, list any other desired qualifications here. Use the specific requirements to differentiate the specific knowledge you need in your particular company from the generic requirements for the position. Derive the specifics from the cultural qualities, preferences, activities, deliverables, and skills identified in the job analysis. An example of a specific requirement might be stated as "two years of machine-vision experience."

Responsibilities: You will need to record the essential job activities, deliverables, cultural qualities, preferences, and skills desired in the position, but the responsibilities section of the template allows you to further refine the requirements to the specific environment. Responsibilities first can be defined by the deliverables in the job analysis; then modified by the essential qualities, preferences, and skills; and finally, the description can be made even more precise by addition of details about desired characteristics, temperament, or personality. For example, you may want a project manager who can get the best effort out of shy-but-talented developers. You may well include "manage projects" as part of the responsibilities requirements, but your awareness of the people already working on the project would merit your also noting something such as, "Must manage projects staffed by shy-but-talented developers."

If the project manager must deal with developers who aren't known for their teamwork abilities, you could say, "Must manage projects, using significant diplomacy with highly skilled, independent-thinking developers who care about the project above all else." Choose your wording carefully while keeping in mind your job description's potential audience. Some of these talented developers may see the job posting or participate on the interview team, and you want them to be happy with the description. Choose your words so as to attract potentially successful candidates without alienating the interview team.

When describing the responsibilities, avoid using jargon. Write short, clear descriptions of the responsibilities. Even if your audience for the job description is internal, your HR staff may not understand your words. Also, if you choose to provide candidates with a job description, jargon may throw them off.

Elimination factors: Use the elimination factors already noted on the job analysis worksheet. Don't forget to consider salary as an elimination factor, but don't list it unless your company practices open-book management. And, if corporate activities, travel, specific-time availability, or a clear, articulate speaking voice are part of the elimination factors, then say so.

Other factors: An optional bit of information to enter on the template pertains to the corporate cultural-fit factors.[1] For a written job description, whether you include cultural-fit factors will depend on what those factors are and whether you want other people to see them. The corporate-fit factors are critically important because they will impact your ability to screen candidates, so discuss them with any recruiters you use, even if you don't write them down. Unfortunately, those factors may be the most controversial inside your organization. Think carefully about which factors you can write down on a document that will be circulated inside your organization.

A True Story

Stu, a VP in a major insurance company, wanted to describe his organization in the following terms: "Bureaucratic organization moves slowly. Change agents need patience." Instead, so as not to offend anyone within the organization by calling it bureaucratic, he called it "a traditional-thinking organization." That more diplomatic wording enabled him to be able to ask appropriate questions and not offend others in his division who read the description.

The Lesson: If you work for a company whose management discourages you from including corporate cultural-fit factors in a job description, don't include them. But, *do* use those fit factors when evaluating candidates.

Use job descriptions to help you screen candidates.

Clearly, job descriptions generally are useful as early screening devices, but not all job descriptions are worth their weight in gold. I've encountered three kinds of job descriptions that are woefully inadequate:

- *laundry lists*—these endlessly detail everything a person would possibly need to know and do in order to succeed at the most senior level for this job.
- *vague, ambiguous, hand-waving descriptions*—these hint at something that might be a job someone wants done, and ignore the personal qualities, preferences, skills, significant activities, and deliverables.

[1] The topic of screening candidates for corporate fit is admirably addressed in Jim Harris and Joan Brannick, *Finding & Keeping Great Employees* (New York: AMA Publications, 1999), pp. 16ff.

- *boilerplate, generic descriptions*—these ignore all the personal qualities, preferences, and skills and assume all people with the title are the same and perform the same work.

A True Story

Following is a verbatim laundry-list job description for a six-month, contract-status, development job. Note that it suffers from the second affliction in the preceding bullet points: It's vague, and it's badly written.

List of Job-Description Attributes

Software Development: Primarily coding in Java.
Development Environment: Java, Tomcat.
Web Services: SOAP, WSDL, XML, Oracle database, PVCS software configuration, management, existing code base is in Visual Basic running on Win 2000 platform.

Systems Understanding:
Ability to understand requirements documents and high-level architecture.
Ability to understand legacy software.
Design/Documentation:
Ability to generate software design specs.
Ability to generate design to support new requirements on an existing system.
Programming:
Excellent programming skills in Java server-side development.
Skilled in database interactions, specifically Oracle.
Web Services:
SOAP—Simple Object Access Protocol.
Generate and work with XML schemas.
Web Services Definition Language—WSDL.
Skills in http data transmission and communications protocols libraries like SAP and socket level programming, and ftp o.
Process:
History of successful software commercialization projects.
Knowledge and proven execution of Software Development processes (e.g. SCM).
Quality:
Track record of quality software development including the creation and execution of unit tests, and release documentation to Software Configuration Management.
Environment:
Demonstrated successful projects involving development and integration across multiple development organizations located in different geographic locations.
Ability to work flexible hours to accommodate multi-national development.
Education:
BS Computer Science + 5 yrs experience.

Interviewer's Notes

Okay, they want Java even though current code is in Visual Basic. Maybe there's a reason farther down.

Ability to understand requirements and high-level architecture is fine. What don't the current people understand?

Hold on. Why XML schemas? How big is this job?

Design specs are good. If the new requirements are for the existing system, who's porting from Visual Basic to Java?

There's a lot more to process than configuration management. Is this job the same as that of release engineer?

The following points stand out for me: Not everyone is in the same location; the work involves mega-hours; position responsible for design, architecture, and release management, maybe porting. All with five years of experience? For a six-month contract position? Uh huh. No one with ten years of experience would touch this job.

> We should add to the laundry-list job description one more requirement: "Able to leap tall buildings in a single bound." Notice that in this real example, which was posted on a job board I saw during a recent training session, there is no mention of deliverables or activities—that is, there is no description of what the person would actually do.

In case you haven't come to this conclusion from what you have just read, let me state my advice unequivocally: Avoid laundry-list job descriptions. Finding someone who meets all those criteria is impossible. Even if you found someone who met the technical criteria, he or she probably wouldn't have your preferred cultural qualities or meet your salary requirements. Unless you're looking for and are willing to wait for and pay a specific, senior-level technical person, a laundry list of technical skills is unrealistic, and only serves to screen out potential candidates. If you've found that you created a laundry-list job description, because you believe that only a superstar can perform the work, ask for help analyzing the work.

Ambiguous, hand-waving job descriptions don't work either—the more ambiguous the job description, the less pre-interview screening you can do. In the laundry-list example printed above, "Process" and "Quality" are part of the description. What about them? What should a developer know about process and quality? How to spell the two words? How to use process and quality to develop more effectively? Which process? Which quality? The job description is too vague to enable a candidate to know whether he or she measures up.

If you have generic job descriptions, even boilerplates, start with them, but don't use them as they are. Customize them with the qualities, preferences, non-technical skills, and specific technical skills you require, so you can more thoroughly screen candidates. You can't easily or successfully recruit, screen résumés, or interview candidates if you use ambiguous job descriptions.

Job descriptions that only address technical skills without any of the cultural qualities are likely to lead you to interview and hire unsuitable or unqualified candidates. If you include typical responsibilities (deliverables and activities) in your job descriptions, you're much more likely to attract candidates with appropriate cultural qualities as well as with needed technical skills.

Identify who will use your job description.

A job description is like any other work document, with the following factors to be considered:

- What audience or audiences are you trying to reach?
- What level of detail do the various audiences need in the job description?

I write job descriptions for internal use only. I don't post job descriptions; I post ads. For me, "internal use" includes external contract recruiters. Because I develop long-term relationships with external recruiters, I treat them the same way I treat the internal interviewers with whom I work. You can get away with using shorthand inside the company and with recruiters who know you and your company well, but be as specific as possible when writing the description. Determine, for example, whether a recruiter needs more or less information than a member of your interview team.

Work hard to specify why you want someone from a particular industry or with technical experience. The more specific you are in stating your requirements, the more successful your screening will be. If you use shorthand such as, "Five years of experience in the medical device industry," you haven't specified your requirements. That experience could mean experience conducting audits, working with patent lawyers, or devising long test cycles, or it could mean something completely different—and it could mean something different to each candidate as well as to you. Be as specific as you can be in the job description so that you attract the people you need.

Instead of using shorthand, try to express your negotiable tradeoffs in order to attract qualified candidates—such as by specifying "Industry experience with 21 CFR Part 11 or experience in other process-audit methods," for example. The more ambiguous you are, the more time you will have to spend working closely with your recruiters, so that they understand what your shorthand means.

You may be working with an HR representative who doesn't know how best to describe technical work, or who doesn't understand your requirements. In that case, your HR rep may want to use some shorthand technique to describe what he or she thinks the requirement means. For example, HR representatives frequently indicate that candidates need a technical degree, because they don't know how to evaluate functional skill and experience. To eliminate this ambiguity, you could provide the representative with a specific description of the requisite qualities, preferences, and non-technical skills *as part of the job description.* Or, you could do what I generally choose—teach the HR rep how to read a résumé with an eye toward detecting requirements.

On the other hand, if your HR rep is bound by some company policy, or honestly doesn't understand how a technical person could be successful without a degree or without experience with a particular operating system, then choose how much energy you want to spend explaining your choices to your HR rep. Don't waste your time, and don't let the HR rep prevent you from finding the best-suited, qualified candidates.

If you work with external recruiters, it's worth taking the time to discuss your personal as well as your organizational assumptions about title, salary, and job responsibilities, so that the recruiters understand what you mean when you talk about a particular job title. Contrary to what many of us believe, job titles are neither standardized nor meaningful from one company to another—they never mean quite the same thing in any two companies. When I use contract recruiters, I explain our job ladder, salary ranges, and how the company works, so they can find appropriate candidates. If you don't develop long-term relationships with recruiters, you'll have to educate each recruiter about your company every time you use one.

For external posting, I write a separate advertisement or use full sentences to convert the bullets in my job description. That way, I minimize the chance that the reader will make incorrect assumptions about what I mean in a job description.

Consider whether your bulleted job description is appropriately worded if you're dealing with someone who wants to publish it for recruiting purposes. It's too easy to jot down vague descriptions or to make an assumption that the reader will understand what you meant (even if it isn't what you wrote) when writing the first draft of a job description.

If the people who will be using the job description are recruiters or interviewers, make sure they know why you're looking for the specified kinds of experience. Use the job description as a starting point for your conversation about this open job.

Learn how best to use standardized job descriptions.

Some organizations do use standardized job descriptions to describe each role on their technical staff: developers, testers, writers, and so on. Standardized job descriptions are a great way to identify what's common among your technical staff members. However, unless you are unique among technical organizations and have fungible staff members who can easily replace each other, you'll need to augment the standardized description with your needs for this particular position.[2] Use your job analysis worksheet to differentiate the candidates you want to attract from those matching the standardized job description. Then, add your changes.

You don't have to use my job description template, especially if your company has its own template. Just be sure you have a place to map all the parts of the job analysis to whatever job description template you use. If you don't, you run the risk of not being able to screen résumés carefully enough, and you'll waste time phone-screening or even interviewing people who are not suitable candidates.

Develop your job description over several drafts.

Don't expect to write a perfect job description the first time. As with any writing, you'll find it easier to write a first draft quickly, put it aside, and modify it later. If you're pressed for time, ask people in your group to review what you have written—and expect to modify and improve the job description while you interview.

As you write ads or phone-screen scripts, you may remember something you want to change in the job description. That's fine—change it. When I start with a new job analysis, I plan on creating two or three drafts of a job description. After I've phone-screened or interviewed a few candidates, I may have more changes for the job description.

If you're having trouble writing the job description, ask the rest of your group, interview team, or the candidate's would-be customers—the people with whom the employee will work—to help you. Your team or customers may be better able to articulate what you're looking for because they are living with the lack of adequate staff every day. You don't have to write a job description by yourself. If you think you could produce a job description that would be more specific or more accurate if you had other people's help, write that job description as part of a team effort. Not only will working with others help you, your interview-team members will have a

[2] The myth of fungible resources is exposed brilliantly in Tom DeMarco, *Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency* (New York: Broadway Books, 2001), pp. 12ff.

better idea of what questions to ask in the interview. And, you'll have a chance to start mentoring your team, in case members of it ever become hiring managers.

Case Study: Walker Software

Dirk Jones has prepared a draft Software Developer job analysis worksheet. Although he is concerned he's spending too much time planning and not enough time interviewing, he's serious about finding the right person quickly, and he's willing to give the job-description exercise a try. At first glance, his template looks fairly complete. His second-draft job analysis worksheet appears below. Immediately following it is Dirk's fleshed-out job description (my comments are noted in italics to show what the description tells me).

Defining Questions	**Answers re: Needs & Observations**
Interaction, roles, level, management component?	Works with developers, testers, writers, project manager, and product manager (or customer representatives) as a software developer. Mid-level, no management component.
Deliverables and activities?	Deliverables: high-level design (post architecture), implementation. Activities: participates in requirements definition meetings, moderates and attends design and code reviews, contributes to smoke test suite, develops integration tests. Leads subsystem development under the guidance of a technical lead.
Essential qualities, preferences, skills?	High collaboration, high teamwork, adaptable, able to consider multiple designs.
Desirable qualities, preferences, skills?	High focus.
Essential technical skills?	C++, UNIX system calls, UNIX shell scripts. Data structures Understanding of telephone industry.
Desirable technical skills?	
Minimum education or training requirements?	Four years working as a developer, at least two completed projects. BS nice, not necessary. BS equivalent okay.
Corporate cultural-fit factors?	Small company, project-oriented, high growth expected, stock options.
Elimination factors?	Salary no higher than mid-range developer.

Software Developer, reporting to Development Manager
Generic requirements:

- Minimum of 4 years as a developer, on at least 2 completed projects
- 2 years of C++
- 2 years of UNIX system calls, data structures
- BS, CS, or equivalent experience. *(Dirk doesn't care whether his candidates have a degree.)*

Specific requirements:

- Understanding of telephony industry. *(Dirk hasn't yet explained why this is necessary or what part of the industry the candidate should understand.)*
- At least 2 years of work on the XYZ subsystem, with responsibility for implementation and unit-testing on completed projects. *(Dirk is looking for someone who's already learned how to design, and who now is learning how to develop the system's architecture. He wants someone who understands how to maintain and add onto code that may not have been sufficiently designed or tested. Note that Dirk hasn't asked for what he wants; he's using this explanation of experience as shorthand for what he really wants. For a first-draft job description, this is okay.)*

Responsibilities:

- For a subsystem: Assist with requirements and overall architecture; design, implement, and unit-test the subsystem
- Moderate and attend design reviews
- Moderate and attend code inspections
- Implement smoke tests for the subsystem
- Develop integration tests for the subsystem
- Work with developers and testers to design, develop, and unit-test the subproject. *(This addresses the high collaboration and high teamwork parts of the essential qualities.)*
- Develop multiple designs for a specific problem. *(Again, part of the essential qualities.)*
- Design for performance and reliability. *(This addresses why Dirk wants someone with telephone-industry experience, so the person already understands the implicit requirements. This extra responsibility wasn't already specified, but was derived from the telephone-industry experience.)*

Additional responsibilities for the appropriate candidate *(Use desirable skills, qualities, and preferences.)*:

- Able to focus on own tasks, even when the rest of the group is working on other tasks. *(Dirk has had problems in the past with developers who couldn't stay focused on their own work, but wanted to solve other people's problems. He's not sure this is the correct way to ask for a person with "high focus" who will mind his or her own business and keep working, but he figures it couldn't hurt.)*

Other factors: None. *(Dirk does not publish salary as an elimination factor. He will find out about salary during the phone-screen.)*

Template 3-1: Dirk's Software Developer Job Description.

Dirk isn't thrilled with how he's described his need for someone with "high focus," but he's written it, and he's clarified why telephone-industry experience is important to him. He can refine the high-focus part later. His job description is still vague as a result of the "understanding of telephone-industry" requirement. In reality, how any candidate understands performance and reliability criteria is critical to product success—and the experience category is appropriate—but Dirk isn't thinking about other industries. That's okay, he'll consider how to ask for what he wants when he writes a phone-screen and develops interview questions.

Now that Dirk has prepared the job analysis worksheet and job description, he can write job postings as well as advertisements.

Points to Remember

- Create a job description that's as specific as you can write it for the position you're trying to fill. If you are trying to fill more than one position of the same kind, list the tradeoffs in the desirable requirements area of the job description template.
- Consider who will use the job description, and write the job description so that person can successfully use it.
- As you write the job description, check to see that you've mapped all the pieces of your analysis to the job description. Decide whether you want to include the elimination factors or company factors as part of the job description, even if the description is just for internal publication.
- Expect to refine, expand, or modify your job analysis when generating the job description. The more you think about a particular problem (that is, the more you think about the reason you're hiring someone), the more you'll understand about how to hire to best fill the opening.

Index

Since 1984.

Dorset House Faxable Order Form • *Fax (212) 727-1044* • *Call (800) 342-6657 or (212) 620-4053*

TITLE ***Save 20% on Forthcoming† Titles***	#	PRICE	TOTAL
Adaptive Software Development (HIGHSMITH)		$44.95	
Adrenaline Junkies (DeMARCO, HRUSCHKA, LISTER, McMENAMIN, J. & S. ROBERTSON) ***NEW***		$35.95	
Agile Software Development in the Large (ECKSTEIN)		$33.95	
Amplifying Your Effectiveness (WEINBERG, BACH & KARTEN)		$24.95	
Are Your Lights On? (GAUSE & WEINBERG)		$13.95	
The Aremac Project (WEINBERG) ***NEW—SCI-FI TECHNO-THRILLER***		$17.95	
Becoming a Technical Leader (WEINBERG)		$29.95	
Best Practices for the Formal Software Testing Process (DRABICK)		$35.95	
Communication Gaps and How to Close Them (KARTEN)		$33.95	
Complete Systems Analysis (J. & S. ROBERTSON)		$57.95	
Creating a Software Engineering Culture (WIEGERS)		$39.95	
Data Model Patterns (HAY)		$39.95	
The Deadline: A Novel About Project Management (DeMARCO)		$24.95	
Designing Quality Databases with IDEF1X Information Models (BRUCE)		$57.95	
Dr. Peeling's Principles of Management (PEELING)		$29.95	
Everyday Heroes of the Quality Movement (GLUCKMAN & ROOME)		$19.95	
Exploring Requirements (GAUSE & WEINBERG)		$44.95	
Five Core Metrics (PUTNAM & MYERS)		$43.95	
General Principles of Systems Design (D. & G. WEINBERG)		$27.95	
Handbook of Walkthroughs, Inspections, and Technical Reviews (FREEDMAN & WEINBERG)		$49.95	
Hiring the Best Knowledge Workers, Techies & Nerds (ROTHMAN)		$37.95	
How to Plan, Develop & Use Information Systems (VAN STEENIS)		$34.95	
An Intro. to General Systems Thinking: Silver Anniversary Edition (WEINBERG)		$33.95	
iTeam: Putting the 'I' Back into Team (PERRY) ***FORTHCOMING***		~~$21.95~~†	**$17.56** before 7/1/08
Just Enough Requirements Management (DAVIS)		$33.95	
Managing Expectations (KARTEN)		$27.95	
Measuring and Managing Performance in Organizations (AUSTIN)		$24.95	
More Secrets of Consulting (WEINBERG)		$33.95	
Object-Oriented Computation in C++ and Java (WEISERT) ***NEW***		$33.95	
The One Minute Methodology (ORR)		$12.95	
Peopleware, 2nd ed. (DeMARCO & LISTER)		$33.95	
Perfect Software (And Other Illusions About Testing) (WEINBERG) ***FORTHCOMING***		~~$23.95~~†	**$19.16** before 7/1/08
Practical Guide to Business Process Reengineering Using IDEF0 (FELDMANN)		$34.95	
Practical Project Management (PAGE-JONES)		$34.95	
Process for System Architecture and Requirements Engineering (HATLEY, HRUSCHKA & PIRBHAI)		$59.95	
Productive Teams: A DVD (DeMARCO & LISTER)		$95.00	
Productivity Sand Traps & Tar Pits (WALSH)		$27.95	
Project Retrospectives (KERTH)		$33.95	
The Psychology of Computer Programming, Silver Anniversary Edition (WEINBERG)		$44.95	
Quality Software Management, Vol. 1: Systems Thinking (WEINBERG)		$41.95	
Quality Software Management, Vol. 2: First-Order Measurement (WEINBERG)		$43.95	
Quality Software Management, Vol. 3: Congruent Action (WEINBERG)		$39.95	
Quality Software Management, Vol. 4: Anticipating Change (WEINBERG)		$44.95	
Rethinking Systems Analysis & Design (WEINBERG)		$27.95	
Roundtable on Project Management (BULLOCK, WEINBERG & BENESH)		$15.95	
Roundtable on Technical Leadership (WEINBERG, BENESH & BULLOCK)		$15.95	
The Secrets of Consulting (WEINBERG)		$29.95	
Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency (DeMarco) **Lower Price!**		$12.95	
Software Endgames (GALEN)		$33.95	
Software Productivity (MILLS)		$25.95	
Software State-of-the-Art (DeMARCO & LISTER)		$45.95	
Strategies for Real-Time System Specification (HATLEY & PIRBHAI)		$49.95	
Surviving the Top Ten Challenges of Software Testing (PERRY & RICE)		$27.95	
Systems Modeling & Requirements Specification Using ECSAM (LAVI & KUDISH)		$47.95	
System Testing with an Attitude (PETSCHENIK)		$39.95	
Testing Dirty Systems (PERRY & RICE) ***FORTHCOMING***		~~$35.95~~†	**$28.76** before 7/1/08
Understanding the Professional Programmer (WEINBERG)		$24.95	
Waltzing with Bears: Managing Risk on Software Projects (DeMARCO & LISTER)		$27.95	
Weinberg on Writing (WEINBERG) ***NEW—THE METHOD THAT BEATS WRITER'S BLOCK***		$24.95	
What Every Programmer Should Know About Object-Oriented Design (PAGE-JONES)		$44.95▲	
Why Does Software Cost So Much? (DeMARCO)		$29.95	
Working Up to Project Management (PHILLIPS) ***FORTHCOMING***		~~$31.95~~†	**$25.56** before 7/1/08

Visit www.dorsethouse.com to Order Via PayPal • No Account Required

†Discounted price valid on orders received before the indicated expiration date. You will not be charged until we're ready to ship. International customers: We'll contact you with shipping options. ▲Qty. limited

SHIPPING AND HANDLING POLICY: First book or DVD, **add $6.00** for UPS shipping. Each additional book, volume, or video up to 5, add $1.25 per item. For 6 or more items, call us for actual charges. Please allow 1 to 4 weeks for delivery. We ship UPS, unless otherwise requested. For UPS, please give complete street address, no postal boxes. First class, air, and all shipments outside continental USA are additional: for estimates, please call, fax, or e-mail with quantities desired, shipping address, and your contact information (phone, fax, e-mail). PAYMENT MUST ACCOMPANY ORDER in US funds; NYS residents, please add sales tax. Prices are effective August 1, 2007, and are subject to change without notice. Online, visit www.dorsethouse.com/shipping.html.

SUBTOTAL:	
SHIPPING:	
NYS TAX:	
TOTAL:	

❑ *Enclosed is my check or money order.* ❑ *Charge the total to my credit card:*

❑ VISA ❑ MC ❑ AMEX Card # ______________________________

Exp. date ____________ Signature ______________________________

Very important → **Daytime phone** ____________________ **fax or e-mail** ____________________

Provide Your E-Mail Address for Order and Shipping Confirmation. We Usually Ship by UPS, But We're Happy to Quote Any Other Service You Wish. ☺

____
NAME TITLE

____
COMPANY

____
STREET (NO P.O. BOXES FOR UPS DELIVERY) ❑ COMMERCIAL ❑ RESIDENTIAL

____
CITY STATE ZIP COUNTRY

DORSET HOUSE PUBLISHING
353 W. 12TH ST.
NEW YORK, NY
10014 USA

(212) 620-4053
(800) 342-6657

Callers, Please Mention "FAX 0508"