

NO- DRAMA PROJECT MANAGEMENT

AVOIDING PREDICTABLE PROBLEMS
FOR PROJECT SUCCESS

BY
BART GERARDI

apress[®]

No-Drama Project Management

Avoiding Predictable Problems for Project Success

Bart Gerardi

Apress®

No-Drama Project Management: Avoiding Predictable Problems for Project Success

Copyright © 2011 by Bart Gerardi

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-4302-3990-1

ISBN-13 (electronic): 978-1-4302-3991-8

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

President and Publisher: Paul Manning

Lead Editor: Jeff Olson

Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh

Coordinating Editor: Jennifer L. Blackwell

Copy Editor: Tiffany Taylor

Compositor: Mary Sudul

Indexer: BIM Indexing & Proofreading Services

Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media, LLC., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

*To my loving wife Judy,
who supports me in all my crazy projects.*

Contents

About the Author	ix
Acknowledgments	xi
Preface	xiii
Chapter 1: The No-Drama Project Manager	1
Chapter 2: Project Management Success	15
Chapter 3: Identify Requirements	29
Chapter 4: Prioritize	45
Chapter 5: Manage Change.....	59
Chapter 6: Align with the Client.....	75
Chapter 7: Testing Assumptions	89
Chapter 8: Identify Decision Makers.....	105
Chapter 9: Communicate Effectively.....	123
Chapter 10: Develop a Plan.....	143
Chapter 11: Prepare for Problems.....	155
Chapter 12: Establish Metrics.....	171
Chapter 13: Know the Roles.....	187
Chapter 14: Handling the Truly Unexpected	205
Chapter 15: The End of Drama	223
Index	241

About the Author

Bart Gerardi is a program manager for an e-commerce company in the E-Commerce 50. He has been a consultant, manager, and leader for 15 years. Bart's love of bringing projects to market has spanned several companies, positions, and waves of Internet fads. Always on the lookout for new projects to run, he also manages the delivery of several teams bringing the next generation of e-commerce to the industry. Bart lives in the Boston area.

You can find more of Bart's writings at www.NoDramaPM.com.

Acknowledgments

There are a lot of people who knowingly or unknowingly made this book possible. These include people who have taught me, people I have learned from, people who worked with me or for me, as well as the people who participated in the creation of the book itself. Without any one of these people, this book would never have existed, and for that I am eternally grateful.

The first group of people are those I've worked for, either in the past or the present. They include Brad Palmer, Adrian Griego, Neil Wheaton, Josh Koppelman, and Tim Martin. These are the folks who felt it was their job to teach me, sometimes accidentally, the practice of running projects. Without them, I wouldn't have had projects to run or experiences to share throughout this book.

The focus of this book is on how a program manager manages project managers. I have to thank my fabulous team for allowing me to manage them for the past several years and practice some of these concepts on them. My team has included Adam Garland, Caleb Munson, Charlene Bunting, Erin DeCesare, Kelli Connors, Matt Gittlitz, and Nicholas Smith, among others throughout the years. In particular, Adam was the one who made me write this book in the first place.

Part of the crux of No-Drama Project Management is working well with your clients. In order for this to work, you need clients who are willing to play along and willing to experiment with you. I've been fortunate to have great clients in Marion Thomas, James Connolly, Matt Guinen, Julia Voorhees, Alison French, Christine Midwood, and my client and thinking partner, Jennifer Hood, plus several score more.

The process of writing a book is a long one, and there are more people involved that you can imagine. This book wouldn't have seen print if not for my great editors, Jennifer Blackwell and Jeff Olson, and outstanding copy-editor Tiffany Taylor, and probably many more who I don't know about. What you hold in your hands is as much due to them as to me.

Finally, the people who gave the most to let me write this book are my family. My wonderful wife Judy, and my three kids, Emily, Brett, and Ben, didn't

mind when I was writing during family vacations, and encouraged me by getting excited when the cover was complete (and sharing with their friends). They were key stakeholders in this entire project, a concept I discuss later in the book.

If any one of the people I've worked with, I've worked for, I've helped support, or who supported me had acted a little bit differently than they did, this book wouldn't exist. It's hard to put into words my thanks, other than to say, thanks for everything.

Preface

If you picked up this book, you are probably a project manager, program manager, or some other project professional. I can tell you, without knowing you personally, that some of your projects have failed. In fact, one might be in the process of failing right now. What you may not know is that not all failures are created equal. Some failures are actually good for your career, whereas some successes may not be. Why is this?

Drama.

Your team, clients, sponsors, and managers want you to be successful. They want every project to create a lot of value and be fondly remembered after it's finished. But this isn't always the case, and it's not always a bad thing. What they don't want is for your project to be full of anxiety and stress. And there is no better way to create stress than by failing in a way that other projects have failed. These include things like lack of preparation, lack of communication, or lack of understanding or alignment on what the project is supposed to deliver. The cause and result of any of these failures is the same:

Drama.

Drama is sand in the gears of any project.

This book explores the predictable ways that projects fail, and how you can work to avoid them. Nobody wants to be on a project that fails for an obvious reason or due to the actions or inactions of the project manager. Projects are inherently risky, and some of those risks are beyond your control. Having one of those risks sink your project isn't necessarily a big deal, nor does it have to be a career buster. But to have an issue that looks to others (and yourself) like a failure of basic project-management skills ruin your delivery may also ruin your prospects of advancement. Fortunately, avoiding that fate is well within your control.

In the chapters that follow, I discuss the problems that doom projects over and over again, and how you can recognize and prevent them. The hope is that if your project is destined to fail, at least it will be for a reason you can be proud of—something you can tell fascinating stories about for years to come.

The No-Drama Project Manager

“We have enough drama around here, without you adding to it...”

People have likened being a project manager to being one part professional bull fighter, one part air-traffic controller, one part toddler daycare director, and one part therapist. It's a job that can be filled with enormous levels of stress, vast amounts of accountability, and often virtually no actual authority. Couple that with the need to make frightening decisions on a nearly daily basis, with limited (and sometimes inaccurate) information, while supporting a team that has as varied and diverse a set of needs as any group imaginable, and you begin to understand what a project manager goes through.

Rarely does the project manager have the authority to fire someone, hire someone, give someone a bonus or pay cut, or even decide which person works on which project. In fact, many project managers have to beg just for the ability to take their team out to lunch once in a while. Yet they're entrusted with project budgets in the millions of dollars in the hopes they can achieve objectives worth hundreds of millions.

Even with this in mind, over 500,000¹ people have received certification from the Project Management Institute (PMI) in the last 40 years, and many

¹ Source: www.pmi.org/About-Us.aspx.

of the practitioners couldn't imagine having any other career. And the people who employ project managers have difficulty figuring out how they'd ever make progress without them. Few jobs are as important, as misunderstood, as difficult, and as maligned as that of project manager.

With this as a backdrop, let's take a look at why we run projects in the first place, and what the roles and responsibilities are for the players involved.

Why We Run Projects

Based on the description of project management, the first question that may come to mind is, "Why would anyone want that job?" In fact, many project managers are used to getting that exact comment from their team, their customers—and even their bosses at times. The phrase, "I don't envy you..." is one that experienced project managers are accustomed to hearing, and they know to shrug it off. If people can't understand why we run projects, there's no way to explain it to them, and it's almost certainly not worth trying.

But there are good reasons. When asked, project managers generally list several explanations as to why they love project management. They usually break down into one of the following three themes:

- You get to learn new things, work with new people, and gain and sharpen skills that you never would have otherwise.
- You're on the leading edge of innovation. If what you were being asked to do was mundane and well understood, you wouldn't be needed.
- You get to launch products and services that the market has never seen before.

The first motivation—the changing nature of the job, the ability to learn from others, and the chance to work with a diverse and fluid team—is one of the most cited reasons why project managers love their jobs. The core skills that make up being a project manager—the ability to plan, to mitigate risk, and to execute—are immutable. However, the business domain, technical challenges, and obstacles to overcome change with each project. You can find yourself solving problems for energy dependence one year, and figuring out how to take e-commerce payments using the barter system the next. And each new assignment brings a new batch of subject-matter experts and team members who are there to help get you to where you want to go.

The second motivation is equally powerful. Some project managers view themselves as trailblazers who create the paths that others follow. There are plenty of jobs for those who want to take what you've done and optimize, support, and maintain it. But while they're off doing that, you're doing whatever is new, shiny, and interesting. Long after what you built is deemed "operational," you've built two or three more things that strive to make the first creation obsolete. You live way out on the edge of safety.

Finally, as a project manager, you get to do something that relatively few other managers get to do—you get to launch. You get to enter a space where there is nothing, and you get to create. Some of the things you build will fizzle into inconsequence, probably before the last cheese puff from the celebration party is eaten. But sometimes, what you build will change the course of an industry or be used by millions who quickly wonder how they ever lived without it. It's this thrill that we get to experience, year in and year out, that our brothers and sisters in delivery don't. Many of them envy project managers and our willingness to launch and watch, rather than have to figure out the details and the total cost of ownership.

■ **Note** Not everything you launch is destined to change the world, but it's the chance that it will that keeps many project managers going.

But before you can take on the mantle of project manager, you must understand what your job is and what it isn't. Not only that, but you may have dozens or hundreds of people on your team, all playing different roles. You need to look at what these people will be doing, how they will interact, and how you can best utilize them on your team.

Why Roles Are Necessary

In football, playing defense involves a concept of *gap assignments*. Those are the areas on the field or in the formation that a specific player is supposed to cover. During the course of the play, it's possible that nothing will happen in that area, leaving that player underutilized and feeling like they didn't contribute to the team, at least for that snap. What the player often doesn't realize is that simply by covering that area, he makes it possible for the coach and the rest of his teammates to focus on other things without having to worry about whether the gaps were covered. Although everyone wants to

directly contribute as often as possible, sometimes allowing the rest of the team to focus on their own contributions is just as valuable.

The same is true when constructing a project team. In a one-person project, you know who is doing everything; if you don't do it, then it's not getting done. In a two-person project, you can make a similar distinction: if you aren't doing something, you can ask your partner if they are. If both of you say no, then you know that no one is doing it. But once a project gets to three or more people, knowing what everyone is responsible for, what they're focusing on, and even more important, what they aren't doing, becomes incredibly important.

Much like the football coach who needs to know that his middle linebacker will indeed be in the middle should the need arise, the project manager needs to know that the engineers are engineering, the analysts are analyzing, and the creatives are creating. Equally important, the amount of effort being spent on each position is an explicit decision. You may choose to have four quality-control people instead of eight, or try to get by with one graphic artist. This team balance is carefully crafted as part of the project plan, and having team members unwind it, by either not knowing what their role is or because you never made it clear to them, will only result in a team that is headed in different directions with different goals in mind.

Because of this, clarity, consistency, and adherence to project roles can become even more important than the people who fill those roles. Better to have your second-string linebacker in the correct position than to have your all-star freelancing and off in the wrong spot—just as it's better to have your business analyst focusing on the problem at hand and not something else that may have come up.

Finally, you may hear comments that project management is easy, or at least not that hard, based on how effortlessly you seem to be able to do it. Take this as a compliment. It isn't a sign that the project is easy or that the role of project manager is simple. What it *is* a sign of is that everyone on the team knows what the projects goals are, knows what their roles and assignments are, and can execute their tasks without anyone being out of position at a critical moment. Crisis management is a real skill, but learning how to avoid a crisis by always having someone at the ready is a much more important and difficult endeavor.

Role of the Project Manager

So, what is the role of the project manager? The project manager is responsible for three completely different functions. For simplicity, let's break them up into the What, the How, and the How It's Going. A good project manager is tasked with executing on all three roles, sometimes separately, sometimes overlapping, sometimes simultaneously. Failure at any one of the three will probably cause the project to fail. Conversely, if all three are done well, you'll get the comments that the project was easy, because everything seemed to go so smoothly.

First, the management of the What. Even if you're blessed with a project that is completely understood, with full and robust requirements and complete alignment among your stakeholder community when the project starts, everything is subject to change, beginning with the first meeting. And you're rarely blessed with such projects to begin with. Later, this book talks about requirements gathering and prioritization, both of which are key components of the What. But even that isn't enough to fully handle this task.

To be a master at gaining alignment around the goals of the project, you need to keep the following questions in mind:

- What problem or opportunity are you trying to address?
- How will this project help address it?
- What are the key successes this project hopes to have?

Keeping these three questions in mind will continue to frame your thinking around creating clarity and alignment about the goals of the project, the actual deliverables the project needs to create, and what the key wins ought to be. The more clear you are on the answers to those three questions, the more clear your team will be, and therefore, the more efficiently your team will operate.

Second is the management of the How. In this role, the project manager is responsible for making certain that the team delivers the project. This means doing regular project-management tasks, such as developing a project plan, managing tasks and performance, and managing the full life-cycle of the delivery. Similar to creating clarity around the project goals and aims, each task and subtask must have equal clarity. This clarity needs to exist not only for the details of the task itself, but also for how it aligns with the overall goal of the project.

Some project managers manage the project team directly, assigning individual tasks to each contributor. Other managers delegate large parts of this

responsibility to team leads or senior members of the project team. Which ever style is used in your environment makes little difference. You're the one who is accountable for getting everything done. The clearer the task description, and the more obvious the alignment between the task and the overall goal, the greater your chances of success.

■ **Note** Project managers often have three different groups they need to manage and keep aligned. In this case, being successful in two out of three areas can be pretty bad.

Finally, the project manager should be accountable for all project communications—the How It's Going. This means mundane things like status reporting and meeting minutes, all the way to interesting things like issue escalation and risk mitigation. The project manager should facilitate collaboration within the team and among groups across teams, as well as drive communications both upward and outward. Everyone will be looking to the project manager for information, and if they aren't getting it or aren't getting enough, or are getting it in a way they don't like, then they will pin that failure on the project itself.

In many projects, communication is most of the work. Preparing team updates, status reports, and weekly reports can take a lot of time. Often, the project manager gets frustrated, thinks “nothing happened this week,” and decides that sending an update isn't necessary. This is almost always the wrong decision. Even if you're right, and every member of the project team was on vacation this week, sending out a “nothing happened” update is considerably preferable to a communication vacuum that leaves people wondering where your project stands. However, it's rarely the case that nothing happened—that's just your perception from being too close to the project. Decisions are always being made, and some progress is always happening somewhere. If you don't let people know, then no one will ever hear about it.

But team updates aren't the only communication tasks that need doing. There are things like steering committee meetings, which can take dozens of hours of preparation for only an hour of discussion. The same is true of executive updates, roadshows, and client presentations. Being able to send the How It's Going message to everyone is a daunting task that can take up a lot of the project manager's time.

Oh: and of course, the manager is responsible for actually delivering on the project. Let's not forget that.

Role of the Project Team

Most project management literature says that the project team is responsible for doing the stuff that the project manager tells them to do. This is, at best, an oversimplification of the role of your project team. Certainly you hope that the members of the team faithfully execute the tasks you put before them; but if that is all they do, then you probably aren't getting as much value from them as you should. And at least some of the time, this is your own fault.

■ **Note** If all your project team does is the things you tell them to do, then you have missed out on a great opportunity.

Yes, as project manager, your job is to fully specify individual tasks that can then be assigned to individual project members for delivery. String enough of these together, and you wind up with a project, a deliverable, or even a product you can bring to market. But there is a major weakness in this kind of thinking. The more you fully specify exactly what you want and exactly how you want it, the less your project team needs to understand the goals of the project and the thinking behind what they're doing. Granted, in certain situations, this is desirable. But in many situations it isn't. You wind up with a much better result at the end of the project if the team understands the goals—understands what you're trying to do—and understands roughly the tasks it will take to deliver on those goals. When you set up this kind of environment, you can sometimes get major over-delivery by your team without them even realizing it.

What you need to expect from your team is a modicum of leadership. This can be specifically team or task leadership, meaning you may be able to delegate individual task assignments to team or group leaders and let them handle the details. Alternately, you can assign very broad-based requirements to these leaders and rely on them to break the requirements down into smaller pieces that can then be managed and delivered by the rest of the team. These leaders probably know their team's strengths and weaknesses, and they probably have a better sense of who should be doing what than you do. If your team has this kind of structure, you would be foolish not to make good use of it and let the team leads help with resource planning and assignment.

But even this won't get the maximum value from your team. This kind of arrangement still requires that all the ideas for what you're building, and all the knowledge about how each piece will eventually add up to a greater whole, resides within a few people. Unless you have a completely outsourced team or no real team at all, this approach won't get you maximum value.

This is why the roles of a high-performing project team are the following:

- Understanding the goals of the project
- Delivering tasks that support the goals of the project
- Suggesting current and future work that further the ultimate goals of the business

When you treat your team members as active participants in reaching the goal, you get a much better result. To make this work, they must know what you're trying to do, and why. With this knowledge, the team can make hundreds of day-to-day decisions correctly without having to ask your help or guidance every time, saving escalations only for when they don't understand or are stuck on a solution.

It's okay if the entire team doesn't act this way. In fact, it's probably preferable. Some of your team members may be on other projects, or they may not be on your project for very long, or maybe they simply aren't that interested in the project in the first place. They don't really want to fully engage with the project—they want to know what their deliverables are and when they're due. A project needs a balance of team members who are committed to the *delivery* of the project and members who are committed to the *success* of the project. Delivery of tasks can be just as important as delivery of value, particularly to a project manager.

Role of the Program Manager

Your program manager, who is also sometimes called a *portfolio manager* or *group project manager*, has many different responsibilities to balance. In some ways, thinking of your program manager as a portfolio manager frames your thinking about him in the proper light. Consider portfolio managers in the financial sense. They're holding a basket of investments, perhaps of several different kinds, and are tasked with maximizing the return on their investments while managing risk. This view reflects an important difference between your program manager and a project manager.

A project manager is responsible for maximizing the return on a project. A program manager is responsible for maximizing the return on all the projects within their portfolio. Sometimes, these two goals conflict. A program manager may decide to under-invest in a project that doesn't have a high return or overinvest in one that does. They may also decide to take a flier on a project that is very risky, with a low likelihood of success, but would have an amazing return if it worked. Additionally, the program manager may decide to have a set of low-investment, low-return projects merely as ballast for the portfolio.

The comparison to an investment portfolio manager is not only accurate, it's the same thinking in just about every way. Projects are individual investments of money, resources, time, or focus. The collection of those individual investments is what your program manager is tasked with maximizing. And just as no investment manager would be evaluated on the success of a single stock within their portfolio, no program manager is evaluated on the return of a single project.

This is an important distinction to make. Although your program manager will be thrilled if your project is an unexpected success, they aren't counting on it. In fact, your project may be near the bottom of the program manager's list in expected return on investment (ROI). Knowing what kind of investment you're managing will greatly change how you run your project. But how do you get your program manager to tell you where you are in the portfolio?

It's not easy. The program manager must treat all projects somewhat equally. If it becomes obvious that someone is running a project with low expectations, then human nature will take hold and the project will live up to its lowered goals. Conversely, if a project has lofty or risky goals, then the manager will feel obligated to live up to those, as well. There is very little upside for the program manager to let the project manager know if they're running a high-risk/high-reward project or one on the lower end of the scale.

■ **Note** Your project exists within a portfolio of other projects. Finding out your place within that portfolio may not only be difficult, it may be counterproductive.

With that in mind, the program manager's role is to get as much return out of the project as possible. Sometimes, this means program managers pay a lot of attention to your project, add resources or help with certain obstacles, or generally look on your project favorably. Sometimes, the opposite

happens. They're fine with supporting your project as long as everything goes correctly. As soon as extra resources need to be applied or anything else happens that changes the ROI calculation, they may decide to pull the plug, not support it, or simply not approve scope or cost increases.

So, as a project manager, what should you do? You need to understand the ROI goals of your project, but you never hear that your goals are low. You also need to understand how your project interacts with the other projects in the portfolio and how it may be a risky project or a risk-mitigation project. Most important, you need to personally track the ROI of your project. If you're about to make a decision, request, or scope change that will affect the expected value of the project, then your program manager needs to know immediately. Even if it seems small or trivial, you need to make them aware of what you're doing. Your program manager needs to maximize the value of all the projects, not just yours, and if they don't have a clear understanding of the return on the project you're currently managing, you make their job impossible to do correctly. And in the long run, they won't like that.

Role of the PMO

In some organizations, there is little distinction between the Project Management Office (PMO) and the program manager. Although this setup may work, the two roles have very different goals and expected outcomes. Your program manager is mostly concerned with the value that the portfolio creates and the short- and long-term ROI of all the efforts going on. Good project-management practices will probably lead to a better project, but most program managers turn a blind eye to less-than-stellar management that drives outstanding results.

Conversely, the PMO should be much more concerned with how a project is run, enforcing standards and best practices, and creating a consistent process for projects. Members of a PMO don't view themselves as at odds with program managers, even if that opinion isn't always reciprocated. Depending on your organization, value may come first, or adherence to standards may take priority. Ideally, you can do both at the same time, but not always.

■ **Note** Best practices should not be at odds with value creation.

I once was the manager of a very large integration with a retail partner that had over 1,500 physical locations. My company was focused entirely on value, the other company entirely on process. At one point, we were asked to create a multi-hundred-page document that would describe the integration to the finest detail—and would never be read by anyone who mattered. However, their PMO would not let the project proceed without this document completed. We toiled on this thing for weeks until we came to a section that we were unwilling to put into the document: details of our own security measures.

Somewhat dejected, I contacted the lead engineer on the other side, asking for his advice on how to proceed. He said something that stuck with me for the rest of the project. He said, “The document must exist, my friend. It doesn’t need to be accurate.” With that wise counsel in mind, the document was completed very quickly. Although it contained no inaccuracies—that wouldn’t be the right thing to do—it also didn’t contain anything useful for anyone who would read it. Luckily, as predicted, no one did.

If your company has a PMO, its reasons for being are standards and consistency. In themselves, these are laudable goals. Your clients want to experience a consistent process each time they engage with you, and the PMO is tasked with sharing knowledge and ideas among all the project managers in the organization in an effort to raise the level of performance of everyone in the function.

Dealing with the PMO is usually easy. Many of the things they want you to do, you would do anyway: things like gathering requirements, gaining sign-off, having a release plan and a risk-mitigation plan, and so on. These are all things that any competent project manager should be doing. Often, the only requirements the PMO enforces involve simple adherence to certain templates, or certain timeframes or tollgates. The retail partner I described had a tollgate that no engineer could be assigned until that document existed. It gave me a good incentive to get the document completed.

The PMO can be your friend as a project manager, but they can be enemies to your project at the same time. When it comes to training, advice, samples, and support, they can be wonderful. If there are new advances or theories in project management, articles or books you should be reading, or classes you should be taking, the PMO is where to get them. But when it comes to running a live project, the project manager needs to walk the balance between what is good for the project and what the PMO expects.

Role of the Project Sponsor and/or Client

Project sponsors or clients share a specific attribute: they're the top-level people who support the project that is going on. This support can be literally money or resources, or it can simply be political capital and vocal support that gets the project approved and keeps it running. It's difficult to run a project without someone in this role—without someone either to champion the project or to pay for its execution. If you find yourself running a project without someone in this role, you should probably recruit someone.

What does this person actually do, though? Your sponsor or client has two very important functions. First, they're the one with final say about any controversial or direction-setting decisions. During the course of the project, you may hit unknown constraints or be faced with choosing one option over another that may change the course of the project. The project manager and business owner—sometimes called the *product manager*—do share responsibility for determining what the options are and, often, coming up with a recommended solution. But sometimes you need to check with the client or sponsor about an issue before going ahead.

The second responsibility of this person is to make introductions and connections. This is generally more true of an internal sponsor than an external client, but it's not unreasonable to expect this from either of them. Your sponsor has a different view of the landscape than you do; rather than focusing on a single project or effort, they should be looking across the company or division to see what else is going on and who the key players are. When your project needs the time of someone outside of the project, whether to get some work done or to provide advice, it's often the sponsor you should turn to for help. Although there is probably nothing stopping you from contacting the person on your own, you may not know the right person to start with or how to approach them. Plus, you may get a better or faster response if the person knows you're coming. This is a benefit not to be undervalued.

In order for sponsors to be effective in this role, they must stay informed about the project. Staying informed can take many different forms. They can read project updates, or they can stay in touch with you or the business owner on an informal basis. They can also sit on steering committees, or they may request that they be updated personally in a different way. Many times, project managers don't see the value in this: they think that having upper management or the client know too much about the project invites meddling or interference. To be fair, this is indeed a risk, but it's a risk worth taking.

Finally, the sponsor will have some impact on your career. If you succeed in a project with a certain sponsor, the likelihood of that person either requesting you or supporting you in the future increases. Additionally, the connections made through your sponsor during the project should endure throughout your project-management career, or at least during your tenure within that environment, and can be used in later projects. In many cases, the people you meet in conjunction with one project turn out to be extremely helpful and supportive in later ones. Of course, being successful in your first interaction goes a long way in turning this into a positive for you.

■ **Note** The value of the project sponsor goes well beyond this one project and can usually impact your career.

When you're setting up your project, identifying who will play this role is a critical step. Sometimes it's obvious—it's the person who is getting the invoice. Other times it's much less clear, such as if you're working on a multidiscipline project or if your project has several possible beneficiaries. Figuring out who to lean on, even if you have to find someone yourself to play the role, will prevent many problems from occurring much later in the project.

Summary

Being a project manager can often be a demanding, thankless task. If you're doing your job well, observers may claim that your job is easy or that the project wasn't much of a challenge. Rarely will they say that it was your own skill in coordination and management that made it look easy. Yet you love the position because you get to do new things, learn new subjects, and launch projects and products into the marketplace that no one has seen before. And the chance that something you work on will make an impact on the company, the industry, or the world is one of the aspects that keeps project managers going and interested in continuing to drag efforts over the finish line.

One of the ways project managers can avoid drama and unnecessary stress is to understand the roles and responsibilities of everyone involved. As a project manager, you're responsible for many tasks, including identifying goals, creating tasks and work plans, and constantly and consistently communicating

how the project is going—to the team, executives and clients, and other members of the organization.

Project team members have the base responsibility of executing the tasks put in their queues, but a handful of them should also be actively engaged with the desired outcome, to help guide the solution and to think of other ideas and possibilities that will help you reach the end.

Your program manager has responsibilities beyond and above your project and is seeking to extract maximum value from the investment placed in your hands. Sometimes, this can mean limiting investment in order to increase the ROI, rather than attempting to increase the return. Similarly, the PMO is responsible for how you run the project, usually without much of an eye to the value you're creating. Learning and spreading best practices, training, and techniques, as well as enforcing some consistency among projects, are key to the success of a PMO.

Finally, the client or sponsor has responsibilities to the project as well as to the project manager. Because sponsors usually have a view of the effort that goes well beyond the project itself, they should be helping with the bigger picture and helping you and the team understand where the project lives within some larger strategy. Also, they should be able to introduce you and the project team to other people and resources who can help, while also helping to remove barriers the project is facing. And your sponsor should be able to help you make relationships that will last much longer than the length of your project and can be used for years to come.

Much of the drama we face as project managers comes from people not knowing or understanding their roles. Generally, when people know what is expected of them, they perform well. When they don't know their expectations, they rarely do well. It's our job to make sure that doesn't happen.

Project Management Success

“What do you mean ‘Expect the expected?’”

A project sometimes fails. Some of the time, it's a spectacular failure. Other times, it merely fizzles out into insignificance. Whatever the case, the project didn't go as planned. Your program manager—sometimes called a portfolio manager or engagement manager—probably won't be all that surprised. Some projects succeed, some projects fail; what matters is that the portfolio is acting as it should. Some projects die out because the idea was bad, or the product is bad, or the market isn't ready. Others encounter wildly unforeseen circumstances, such as a breakthrough technology that renders a product obsolete nearly immediately, or a crisis or legislation that makes bringing something to market impossible. Unforeseen, unmitigable problems cause portfolio managers to shrug, curse a little, and then cut their losses. No sense throwing good money after bad.

■ **Note** If projects always went as planned, project managers wouldn't be needed.

But if you want to frustrate a portfolio manager and have them muttering that they should have been a farmer, the easiest way to do it is to fail for a reason that was expected. That is, the project itself was fine—it was the management of it that caused it to fail. In other words: you. Hundreds of thousands of projects are currently being managed by nearly as many people, who identify themselves as project managers. Of those projects that get cut short, die off, or otherwise go to the project graveyard, a significant portion don't succeed for reasons that could have been predicted before the project began. These include the usual suspects, such as communication and risk management, as well as a few more items learned only through hindsight, like identifying key decision makers or waiting to unleash your development team until you're ready to manage them.

Your goal as a project manager is not to make every project you manage a rousing success. That's overly optimistic. Your goal should be to let your mistakes be chalked up to novel circumstances or something unknown. Your portfolio manager has in the back of their head a list of what former United States Defense Secretary Don Rumsfeld called "known unknowns." Having your project run afoul of one of them won't be greeted with a smile and pat on the head. Sure, you didn't know where the problem would come from, what form it would take, or when it could come; this is true. But you knew, or should have known, that it would come eventually. Failing to be ready for it—or failing to actively manage it—is just as bad as having your dependency chart be missing a step, or having your Gantt chart not line up. That is to say, it's not okay at all.

You should strive to make your projects succeed or, if not, to catch fire in some very interesting way. A project that misses the mark in a predictable way makes no one happy, your manager least of all. Learning how to identify and avoid some of these "known unknowns" is what this book is about. This way, at the very least, your project post-mortems can be full of new and educational stories, not just the same list of bullets from your last project with nothing but the headers changed.

When project managers get together to swap stories, the good ones aren't those that begin, "Remember that project where the communication was poor?" That doesn't really narrow it down, plus that isn't the making of an interesting story. It's a problem we've all faced and will face again in the future, so no one wants to hear you recount it. They want to hear something they've never heard before, or something they can learn a lesson from. Even more preferable would be both.

The following sections offer a rundown of the known unknowns that no-drama project managers keep in mind. Each topic is developed further in the book.

How to Irritate Your Program Manager

Having fun stories to tell your colleagues is an interesting motivator for success. But an even better one is making sure your next assignment is a good one, and not a project that no one seems to care about.

There is no quicker way to find yourself back in the minors than doing things that irritate your program manager. Having your project become a failure isn't enough to do this; many projects fail despite the best efforts of the team. In order to really get on your program manager's bad side, you need to have your project fail either because you didn't anticipate something obvious, or because you did something that wound up creating a headache or problem for your program manager to have to deal with.

Failing at the Obvious

Your program manager knows that not all projects will be successful. If it were that easy to have a great idea and run the project perfectly, then your job wouldn't be nearly the challenge that it is. But the way to really bother your program manager is to have a project fail thanks to mistakes made by you or your team. Your program manager is more than willing to write off a project if it turns out to be too difficult, or the landscape shifts, or it's just a bad idea. What they can't dismiss is you failing to plan properly, or a foreseeable problem that takes you completely unaware.

The thing is, you can be running your project without issue, meeting all your deliverables, and keeping everything seemingly on track. If you use a stop-light-type dashboard, you may be reporting your project as "green" every week or month. Everything seems to be working fine, and you have no issues to report to the client or to your program manager. Your plan is to carry on doing what you've been doing. And then, something changes. It could be that client isn't aligned with what your project is producing, and possibly never was. Maybe the person you went to for validation wasn't authorized or skilled enough to grant you the validation on which you were basing your progress. Possibly you were doing really well on the low-priority items, but you hit your first snag on a higher-priority one. Whatever the cause, the client no longer believes you when you say everything is on track.

Alternately, it could be a problem in the way you're running the team. Maybe they thought they were doing the right thing, but it turns out they misunderstood you. Or maybe your plan wasn't resilient enough to withstand an otherwise fairly benign change. Perhaps you're doing well against the metrics you're using, but those aren't the measurements your client or your program manager is using. In addition to your client disbelieving that your project is going well, perhaps now the team is thinking the same way. And this from a project that was doing so well just recently.

Being a Headache

A friend of mine often says that being able to run a project is table stakes for being a senior project manager. It seems like a silly or obvious statement, but his meaning is clear: if I can't trust you to run the projects I assign to you, then you can't be on my team. Of course, project managers have all levels of skill and experience, and it's the program manager's job to make sure the project difficulty matches the capabilities of the project manager. But if you're given a project that you should be able to handle, then you should be able to manage it without causing your program manager headaches.

■ **Note** One of your primary tasks is to reduce stress on your program manager, not increase it.

There are few better ways to create headaches for your program manager than out-of-channel escalations. These are direct contacts with the program manager from someone on the team or someone on the client side who doesn't think that the project is being run correctly. This isn't you hitting a problem that you need help solving or a decision for which you need the program manager's approval. It's not even you going to the program manager to admit a mistake or alert them about something that's coming. You expect those situations, and those are actions you expect you to take.

The headache comes when the initiator of one of these conversations is someone other than the project manager. For instance, the client may believe there is a failure in communication, or someone on the team may think there is a flaw in the plan. Perhaps another manager witnessed something they think needs more oversight, or someone saw some bad behavior and wanted to report it. Whatever it was, it has caused stress for the program manager.

This kind of escalation can cause a lot of work for a program manager. They first need to ascertain whether the topic being discussed is worthy of being escalated. Next, they need to determine if the escalation is true, partially true, or not true. Then, depending on the veracity and severity of the incident, the program manager probably needs to act on the information. This is a lot of work, and work they'd rather not be doing; they hired you to run the project, after all.

Still, this situation may not be all bad. It's possible that something happened while you were away or doing something else. You can't be everywhere. Or perhaps someone knows something you don't, but they're unsure how to get you the information. Or maybe the issue is something so innocent or minor that it doesn't wind up causing much work. What is bad is when someone escalates something that should be foundational to the project. When a program manager discovers that basic items are being missed, they realize that they have a project manager they can't trust. And that's going to cause them headaches.

■ **Note** The level of drama in a program manager's portfolio is directly related to how much they can trust the project managers on their team.

A project manager's job consists of two overlapping but fairly distinct areas: client management and team management. Expected problems can come from either vector or sometimes from both at the same time. What follows in this chapter and the rest of the book is an exploration of some of these issues and why you, as a project manager, need to focus on making sure the problems you bring to your program manager aren't on this list.

Expected Problems in Client Management

Many times, client management isn't about giving clients what they want. Often it's about convincing them that they want what they're getting. One of your roles as a project manager is to guide the team toward delivering what the client is asking for while simultaneously guiding the client toward what the team is delivering. Managing this middle ground is tricky, but it's vital to creating a satisfied client.

However, there are some sure-fire ways to create a dissatisfied client that every project manager has lived through at one point or another. You want

to avoid them. Although it's possible that you'll eventually disappoint, it's often more acceptable to do so in an unexpected way rather than in one of the standard ways projects tend to go awry. From your program manager's point of view, the following failures are a lot less acceptable than finding a new way to fail.

Requirements

A good project manager knows that you need to gather requirements from the customer. You can't have a project unless you have requirements. So, many project managers are diligent about the process, and they schedule a lot of time with customers who tell them everything they want done. The project manager, again being a diligent worker, takes all the things they hear and creates a requirements document. This document is approved by the client, contains specific and actionable items, and is used as the primary source for the project going forward.

It may also completely miss the point.

One of the things your program manager does not want to hear is that the requirements you were building were signed off on, that won't rescue you. Obviously, if you are dealing with a contractual or governmental obligation, then having your document approved is a very good thing. But if your goal is client satisfaction or project success, then having a document that details each specific task for the entire project doesn't help you much. A project isn't just a collection of individual tasks, or at least it shouldn't be. A project should be greater than the sum of its parts.

Although there is a lot of comfort in figuring out what those parts are, writing them down, and having someone deliver them, you may be missing where you need to add value. There is a question that can help you focus your efforts and those of the client toward the goals of the project, and not just specific tasks: "What are we actually trying to do with this project?" If during or near the end of the project, you tell your program manager that you delivered everything the client wanted but the project is a failure, then your program manager will start to doubt your ability to gather the real requirements. Spending too much time on low- or no-value items is just as bad as spending no time on the project at all.

Prioritization

Similar to gathering requirements, you must also help the client and the project team focus on prioritization. If you aren't paying close enough attention, you may wind up spending a lot of time on requirements that turn out to be not valuable and not critical to the success of the project. You may find that you're working on a lot of tasks, and probably getting a lot of them done, but you aren't any nearer to your goal. This becomes motion without progress, which looks good for a little while but gets you nowhere.

A good prioritization exercise helps the project team and the client discover where the expected value of the project lies, and how you plan to address it. Without doing this exercise, you have only a vague notion of what clients think they're trying to accomplish; and, at best, you can only hope to deliver things that are related to that goal. A highly skilled project manager can probably accumulate enough of these kinds of tasks to eventually create something that drives significant value for the client. But it shouldn't take a vast amount of project skill to find out what the client is actually trying to do. Getting your arms around the project's primary goals should be job number one.

And here again you hit what a no-drama project manager needs to be prepared to handle. Being able to focus the client and the team on the top priorities is an important way to limit indecision and confusion in your project. Conversely, if the project manager lets the team spend significant resources on lower-priority tasks, no matter how compelling they sound, then they're asking for trouble. Even worse is telling your program manager that, due to expending too much time and effort on lower-value items, you now need to extend or expand your project budget. Without question, this line of reasoning will get you in hot water with your program manager, and probably your client as well.

Change

Change in your project is inevitable; in fact, it's probably welcome, because change is normally the result of a refinement in the idea or the goals of the project. However, in order to reduce the level of anxiety your team feels, you need to be sure you're making changes in a transparent and intentional way. The change, whatever it is, will impact the people, the project, and the situation in the organization. It's important to treat this impact with respect. Making a change assertively, rather than allowing it to happen on its own, is an important step to managing the change effectively.

■ **Note** The theme of making intentional decisions is one you hear throughout this book.

After a change happens, it's important to look back on what remains, to see if the project is still recognizable. If the focus or goals of a project have altered drastically enough, then trying to continue with the project as currently constructed may be counterproductive. You should confirm that the project is still worth doing and that your current approach is adequate to deliver it.

Doing neither of those things is a way to invite drama into your project. If you're facing change but ignoring it, hoping that it will go away or manage itself, then you leave your team feeling undirected and confused. They will notice and wonder why you aren't doing anything. And if, after the change, you're left with a potentially disappointing deliverable, your client will wonder why you aren't acting on it, either.

Alignment

You need to adhere to two different types of alignment. The first is alignment of your project to corporate or organizational strategy. You should be very aware of where your project exists in the overall strategy of the company or client. What was once a vital part of the strategy may change over time, even if your project doesn't. Although it's beneficial for many reasons to work on a project that is in perfect alignment with the corporate objectives, that isn't always the case. But knowing where the project lives in the corporate ecosystem is important.

Second is the alignment of your stakeholders with the project's goals. Even if your project is well aligned with the overall desires of the organization, if your clients, stakeholders, and team aren't aligned with the goals of the project, then you won't get anywhere. Like a shopping cart with one wheel that isn't pointing in the right direction, one element pulling in a slightly different direction can cause a very large negative impact on your ability to deliver.

Misalignment is another primary source of drama in projects. If you're running a project that is misaligned with organizational strategy and vision, then people will question why you're doing it at all. And if team members or stakeholders don't agree about what the project is trying to accomplish, the result is friction within the project that has to be overcome every time you try to make progress. Both levels of alignment are vital to a well-functioning project.

Assumptions

Projects, like life, run on assumptions. Without them you can't make progress, because you would wait until you had perfect and complete information before making any decisions. As it happens, most assumptions either turn out to be true or turn out not to matter as much as you once thought. For these reasons, you make assumptions about projects all the time, and you'll often find that your projects benefit from it.

■ **Note** You make assumptions frequently, and your life and projects are better for it.

However, when you make an assumption, you need to be clear that you're making one. It's easy to base a project on something you believe to be a fact, when it's actually only something that you're assuming to be true. Assumptions can be difficult to spot, because often they appear to be assertions of fact, and sometimes they come from people who are expert enough to know the difference. Also, discovering how much confidence you have in an assumption, and whether it's more of a guess or wish, is vital to creating a solid project. Constant vigilance over your foundational data is required of a project manager who needs to understand that data.

When you've identified your project assumptions, you need to constantly validate, update, and test them. Most of them will turn out to be true and can eventually be retired. But some of them will wind up either not being true or being less true than you'd like. These become full-fledged risks, and they must be managed like other risks. Just because you base your project on an assumption doesn't give you license to ignore it and hope it eventually turns out okay.

Identifying Stakeholders

Many a project has gone wrong because the project manager improperly identified decision makers. In many ways, basing your project on decisions made by the wrong people is worse than making no decision at all. Your team may have been confidently headed off in a direction, only to find that the direction isn't correct, or that the person who made the decision had no authority to make it, correct or not. This can be a little deflating, especially if it takes you a while to figure out who can validate or approve the decision and who actually has final say.

Then there are the stakeholders who aren't in the direct line of the project. These people have indirect influence on the project, either because they have something the project needs, such as access to a key resource or information, or because they have the trust of the people who are making the final decisions. The more that you, as project manager, are able to correct identify these kinds of stakeholders, the more you can make sure all the key decision makers, both direct and indirect, have a voice in the project and make decisions that stick.

Expected Problems in Team Management

As mentioned, the role of project manager lies at the intersection of client management and team management. Although many things can go wrong in your handling of the client, many of them are eventually reversible. Relationships tend to be malleable, and what looked like trouble one day can become a strength the next. Your clients' opinion of you can change for the better overnight, and the past has little bearing on the future.

When it comes to team management, that isn't always the case. It takes longer to recover from mistakes, and getting over a loss of trust can take even longer. This effect is compounded when the error was foreseeable, or the team knew about it and figured that you knew about it as well. Fortunately, the converse is also true. If you make a wise decision or make a difficult decision correctly, you may be able to win over the team enough so that the next time you have difficulty, they're more inclined to follow your lead.

The following issues are ones that all project managers face and have to overcome. Some of them may seem obvious, whereas others don't seem like that big of a deal. Probably both viewpoints are true. But without a good understanding of the foundation of team management, you can't hope to do a good job managing your team or managing upward when your team is having a problem. Unless you perform well in these areas, the rest of your team-management skills are unlikely to matter.

Communication

Communication is the means by which project managers can lead and grow their teams, manage their clients, and run projects. In fact, communication is the only way to do these things. Many things contribute to effective communication, including form, content, and medium. The only way you can decide on a communication style and method is to understand your message

and your audience. Your team, your client, and your management differ in what they want to hear and how they want to hear it.

Treating your communications as deliverables will assist you greatly in making this kind of determination. For a large portion of your project engagement, communication is the only thing you can provide the client. Until you get to the stage where you have a prototype or other tangible items to show, your communication ability is the way the project community can judge how a project is going. Communication needs to be considered an important activity in its own right, not something that gets done when you have time.

This means gathering requirements. It also means determining how people want to get information, what they want to get, and how often they want to get it. It means testing your messages for effectiveness by asking people if they find what you're providing to be useful and adequate. And it means managing change: just as the needs of a project can change along the way, so too can the needs of your audience. Being sensitive to this and adapting is as important as managing the changing nature of your project.

Planning

Planning is something that all project managers must do—and must do well to be successful. There is a tremendous amount of value in knowing where you're headed before you begin; if you start without a plan, you have no idea what you should do next. Project managers usually have to make several kinds of plans, from project schedules to work-breakdown structures, risk-mitigation plans to meeting plans, contingency plans to rollout plans. In fact, some project managers create plans of plans, because there are so many to keep straight.

Plans are certainly important. Without them, you have no means to plot out your progress and track how you're doing against your goal. However, it's possible to fall into the trap of managing the plan rather than managing the project. Plans exist to aid in the project, but they aren't the deliverable. If you follow your plan perfectly, but you produce something substandard, you can't get away with claiming victory. Project management is a blend of creating and executing plans, and being flexible enough to adjust when the time comes.

Preparing

Often, a project manager's ability to prepare sets high performers apart from others. Preparing is similar to planning: you think about what to do in

the future. However, planning is done based on things you know will happen: you know certain tasks need to be completed, you know when people will be on vacation, and you know the dependencies and ideal sequencing of tasks ahead of you. Preparation is the opposite: you're planning for things that you don't know will happen, but that you do know will affect the project if they arise.

This process goes under many names, such as risk-mitigation planning, scenario planning, and "what if" analysis. The outcome isn't a concrete plan that drives the solution forward. Rather, the desired outcome is a flexible set of plans that can be ready just in case something happens that greatly impacts the project. It can be very difficult to be ready for any possible occurrence. There simply are too many possibilities to try to be fully prepared for each of them. However, there are usually a finite number of impacts that you need to be prepared to handle. And that's what the goal of the project manager should be.

■ **Note** Planning is for things you know about. Preparing is for things you don't.

Once you have these kinds of preparations created, the final piece of the puzzle is recognizing when you can use them. Challenges and opportunities present themselves in all manner of different ways, and it can be very easy to think that you haven't thought of this specific scenario. It's probably true that you haven't, but you've thought of one that's close enough that the solution remains the same. Recognizing this is one of the keys to being a successful project manager.

Establishing Metrics

Knowing what you're trying to accomplish is important in all walks of life. If you don't know what your goals are, you won't know when and if you've reached them. However, relatively few of life's or business's goals are black and white. It's fairly rare that you're either a complete success or a complete failure. This is where the concept of Key Performance Indicators (KPIs) comes in. During the process of your project, you likely have several KPIs that you're trying to improve. Some of them will be more successful than others, and some of them may go down in the service of making others go up. This is normal.

But if you don't do the analysis ahead of time to determine what the KPIs are and what changes you hope to achieve, you don't know how to make these

kinds of decisions intentionally. You need to know what you're measuring, how you plan to do the measuring, and what change you think is acceptable.

■ **Note** Without KPIs, you don't know how well you're doing.

You can also use metrics to measure the project. This helps answer the question, "How am I doing as a project manager?" It's possible that a dismal failure of a project is run fairly well. It's also possible to have a rousing success be successful mostly in spite of the work of the project manager. If all you do is look at the project's output, you don't get a full picture of the project manager's performance.

Knowing how the project is measured is key to determining whether it's a success—especially in an environment where to make something better, you may need to make something worse. Being able to measure change is important to crafting a final story. Equally important is being able to measure your own performance and not conflate the results of the project with the skill of the project manager. Having a well-reasoned set of KPIs can help with both needs and is the primary way for you to declare victory or suffer defeat.

Project Roles

On a high-performing team, everyone knows what position they're playing. This is true of all kinds of teams, from sports teams to project teams. In the project context, just as in sports, people can play two different kinds of roles based on where they're situated and how much influence they're supposed to wield. Sports teams name captains, who are usually obvious because they tend to have the letter *C* embroidered on their uniform. Project teams don't generally go that far, but the concept is the same. Some people on the team are expected to be leaders, and others aren't. If someone is unclear which they are, they're more likely to cause disruption than to behave correctly.

Like many other things in project management, this is an expectation-setting task. People need to know what position they're playing, so they know which tasks they should be prioritizing and what they're being counted on to produce. If team members don't know what skills and talents they're expected to use while on a project, they can only guess the right way to work. Guesses are sometimes correct, but if so, it's usually by accident. And you need to limit these kinds of accidents in your effort to limit drama.

Additionally, your leaders need to know that they're expected to lead. It sounds simple, but it can be easy for leaders to slip into being followers, either causing a void where you need someone or necessitating someone else stepping into the role. A great way to cause confusion in your project is to expect people to set direction who don't know they're supposed to do that or are unwilling to do so. That's a problem you need to correct immediately.

Not All Problems Are Expected

It would be unfair to expect a project manager to anticipate every single thing that can possibly go wrong. But you do need to do a little investigation before you declare a problem completely unforeseeable. Before you assert that something that has come up is a complete surprise, be sure it's true. If you attempt to blame problems on unforeseen issues, and it turns out that you might have been able to prepare for those issues, you lose a lot of credibility with your team, your client, and your program manager.

That said, when something does come up that you haven't prepared for, you have several options. Each option has different outcomes, pros, and cons, so choosing carefully is important. This is where your relationship with the team and client come into play. If you wisely decide how to handle these issues, you can set the project on a good path to success. If your team and client don't trust your judgment due to other issues, you can wind up going in the wrong direction and find it harder and harder to recover.

No-Drama Project Management

The rest of this book contains deep dives into each of the topics described in this chapter. Each subject is nuanced and particular, and each has the potential to cause significant drama in your project if not handled correctly. As a project manager, you should be willing to deliver value to your clients, enable their success, and launch impactful new projects into the marketplace. But at the top level, you're trying to build a long-term history of successful implementations and client engagements.

As you hear later in the book, this breaks down into two questions:

- Does the team want to work with you again?
- Does the client want to work with you again?

You can be sure that the answers to these questions are directly related to the amount of drama you create and invite into your projects.

Identify Requirements

“Tell me again, what are we trying to do here?”

It feels somewhat obvious to say that knowing what you hope to accomplish within the confines of your project is the very first step toward success. However, it isn't uncommon for projects to get well on their way without really knowing what they're trying to do. Some projects even manage to launch without ever figuring this out. Most such projects don't do well once they hit the marketplace, if they're able to find a market at all.

In *Alice in Wonderland*, Alice asks the Cheshire Cat, “Which way should I go from here?” The Cat replies, “That depends a good deal on where you want to get to.” When Alice confesses to not knowing where she wants to wind up, the Cat tells her that it then doesn't matter which way she goes. This kind of thinking is unfortunately found in many organizations, companies, and projects. They don't have an agreed-upon, or even a well-understood, destination, but they're merrily marching along a road to somewhere.

Simply having a dutifully well-documented list of requirements isn't enough, however. You've probably seen examples of requirements documents that weren't all that helpful, or were the opposite of helpful. For instance, a document may be specific down to the finest detail (such as how many pixels wide the logo on a web site is supposed to be) but unclear or silent about who will be using the web site. Or it may reference style guides that talk about which shade of red to use, or which font size and face to display in, but have a section titled “Twitter Integration Goes Here.”

If you're handed a requirements document at the beginning of your project, don't be fooled into counting pages and using that number to determine the document's completeness. The document's contents may be true and accurate, but that doesn't mean it contains everything you need to know. On the contrary, people tend to do the things they understand well at the start of a project, leaving the unknown until later. This is a valid strategy, because as time passes you learn more about those unknowns. But there will come a time when you have a document that does an excellent job telling the reader what everyone already knows and makes no mention of the things that are truly difficult or interesting.

Note People tend to document the things they already know, not the things that they haven't figured out yet.

The worst of all offenses comes when a requirements-gathering effort focuses too much on system and tactical requirements and not enough on what you're actually trying to accomplish. Many millions of project hours have been spent building features that seemed interesting to someone at the time but were never used by customers or clients. Similarly, teams spend a lot of time building systems that support processes that are either obsolete by the time the system is ready or, worse, help speed up a process they hoped to retire.

There are many ways to avoid this kind of requirements-gathering. Most of them focus on either use case-based requirements or user stories, depending on the methodology used. Dozens of fabulous books have been written about how to use these techniques in drafting your requirements, so this book doesn't cover them. But the principle behind use cases and user stories is important. They view the system you're building from the user's eyes and determine exactly what the user is trying to do while interacting with the project. Sometimes the goal is to do things better, sometimes it means faster or more efficient, but sometimes it can mean making connections or inferences that otherwise would go unnoticed. The point of these methodologies is simple: if you don't know what your users are trying to do, then you have little hope of building something useful for them.

This chapter is a microcosm of the essence of *No-Drama Project Management*. The key things to focus on are determining why you're doing what you're doing, figuring out who the players are and how they interact, avoiding getting

hung up on details that don't matter, and thinking about things through the eyes of your customers. If all you remember are those four principles, then you'll understand the rest of this book.

Customer Identification

The very first part of requirements-gathering is figuring out who the project's customers are. Often, the customers and the users are the same people, but not always. *Customers* are the people for whom you're doing the work. A customer can be a division head, an executive, or an outside client or company. Customers can also be people important in the organization, no matter what title or role they currently carry. All projects have a short list of people who can determine whether the project was a success. You need to identify these people.

Once compiled, the list may surprise you. Although it will contain many people you expected, there may also be people on the list whom you didn't realize carried so much influence. They may be important due to their tenure, role, or subject-matter expertise, or merely because they have significant respect within the organization. Figuring out who these people are and determining their keys to success is vital to your project. In Chapter Eight, we will talk about Stakeholder Analysis, and how people that seem to be unrelated to a project can have a large impact in determining whether or not it is successful.

Additionally, some people are negative indicators. That is, they can't tell you if you did the right thing, but they can certainly tell you—loudly—if you did the wrong thing. This list can include people who do this kind of activity for a living, such as those in legal, compliance, and audit, as well as individuals who are simply resistant to your project. In some cases, these people specify requirements that are true project constraints; your project will be deemed a failure if it isn't compliant with regulations, standards, or whatever. In other cases, these requirements are land mines that you need to navigate throughout the course of your project. Someone may say they're must-haves, but that doesn't necessarily make it so.

The customer list now includes four types of people:

- The people who are supporting or paying for your work
- The people who will use the outcome of your work
- The gatekeepers, who make sure your work is in compliance
- The obstructionists, who have objections, real or imagined

To a degree, all of them are important, but some more than others. But without identifying your constituent base and deciding which of the categories a specific person belong in, it becomes a serious challenge to manage their expectations and elicit their requirements in any meaningful way. This is true in both directions: you can't allow the gatekeepers to determine what the functional requirements should be, nor can you allow your end users to decide what would pass compliance and what wouldn't.

Note All projects have constituents. Understanding these groups will help you gather better requirements.

How do you find out who is in which group? You simply ask them about their interest in the project. You should be able to tell which people are interested in the solution to a problem or the opportunity the project presents. Equally, you should be able to spot the people who have little excitement about the project itself—as long as it doesn't mess up end-of-quarter reporting. And finally, you should find the people who are entirely negative about the project even though they don't seem to have any real stake in the outcome.

Only now are you ready to start interviewing for requirements. If you begin without taking the time to identify your sources and identify your customer groups, then you don't have the context to understand what you're gathering. In fact, you may find yourself in front of a gatekeeper, beginning a sentence with, "But I was told that..." never realizing that that person who gave you compliance requirements wasn't in a position (or didn't have the knowledge) to do so. Don't fall into this trap.

Like almost everyone, I have a fairly embarrassing story of my own to go along with this advice. Long ago, I was working on a new payroll system for our HRIS department. Everyone was working under the assumption that July 1—the first day of the fiscal year—was an important date on the calendar. All paychecks turned over on that date, all expense reimbursements were due, and all sorts of activity occurred on that day. This date requirement (which was worded as a constraint) seemed incredibly real. Everyone I spoke with was certain about the importance of the beginning of the year and that if we messed with it, we would have to start over.

Our solution worked with this information as if it were immutable. We had a long series of flow charts and sequence diagrams, and we created a very elaborate, elegant design that allowed for all processing to take place within

that short window of time but let the turning of the new year happen without incident. It took quite a bit of extra work, and we had to cut out some features in order to make it fit, but the business owners agreed that it was worth it because the constraint was so important.

Except that it wasn't. Everything happened on July 1 just out of convenience. Back when people worked with paper, pencils, and ledger books, it was much simpler for the bookkeepers to perform many tasks on one day while the information was pulled out and available. When they switched to the miracle tool that is Excel, around 15 years ago, there was no longer a need for all the activities to happen on July 1—they just saw no reason to change. If we needed to swap the date around to make the project easier, we should have gone ahead and done it.

Of course, I learned this well after the cost had been paid. The designs were finished, and most of the work was completed. When I informed the business owner of the project that the constraint we were working under wasn't real, his only response was, "Well, I'll be darned. Oh well." And then he instantly forgot about it. The key constraint that had kept my team working for months was for nothing. And all because I didn't take the time to realize who I was talking with. I was taking financial gate-keeping requirements from HR, and I shouldn't have been shocked to learn that they were wrong. The fact that everyone in HR told me the same thing didn't make their misconception any more true.

To this day, as far as I know, the company still does everything on July 1. I don't know whether they're still using the systems that my team built years ago, but I suspect if the systems were redone, some other poor project manager heard the same advice I did: "Don't mess with the magic day." And like me, they probably listened.

Top-Down Thinking vs. Bottom-Up Thinking

Relatively few things make me more nervous about a project than what I'll call *bottom-up requirements gathering*. These projects can also be called *laundry list projects* or *aggregation projects*. You've probably been faced with running these kinds of efforts. A business owner, a group, or an executive has a bundle of unused resources but no idea what to do with them. Instead of coming up with a good idea, they create a project out of the wish list of everyone on the team, department, or division. Because the project needs to be managed and coordinated, it needs a project manager. And that's you.

As a program manager, I'm always reluctant to start these kinds of projects, preferring to break them into smaller projects if possible. However, sometimes these kinds of projects happen organically. It's possible that a project began with a well-formed idea and a fairly clear objective. But as you start collecting requirements and talking with people about the problem domain, you learn more things that you really should be doing. Usually, you're right—those things ought to be done. They just shouldn't be done as a part of your project.

Note Wish-list projects can happen—and grow—organically if you let them.

This kind of project, where you talk with a lot of people and come up with a bunch of good (if disconnected) ideas, isn't a project at all. You may call it that, and you may have a project kickoff and team meetings, and actively manage all the people assigned to you. But without a clear, short list of objectives and key success criteria, and a very small number of decision makers, you don't have a project. You have a mess.

A much better approach is to operate top-down. Of course, in order to do this, you must identify who the top is. It may be a single executive, the manager of a department, or someone who is playing the role of client or customer. Focus on what this person wants done, and distill that information into a very short list of key success criteria. A list of key objectives with only one item on it isn't a bad thing and in some cases may be preferable. Everything you gather that doesn't line up with the success criteria should be ruthlessly cast overboard, no matter how objectively good the idea seems to be. That's not to say that these aren't good ideas—they probably are. You should document them and prepare to use them in a future project.

You need to continually ask the owner of the project, "What are we trying to do here?" Only accept an answer that can fit on one side of a 3 × 5-inch index card. If you don't have that, try again. Consider using an actual card if it helps.

Why is this important? One of the key points in John Gray's book *Men Are from Mars, Women Are from Venus*, is the idea of people using a different scoring system when valuing acts and actions. One side believes that as long as you deliver the very top priority, even if that is all you deliver, that makes up for missing everything else. The other side believes that everything on

the list is of equal value, and as long as you deliver enough of it, they're pleased with the outcome.¹ This is a problem you're going to face.

The top-down method is the former type. You should work on requirements, features, and capabilities in strict priority order. There is no sense in working on priority number seven if priority number one isn't done and is at risk. Many prioritization strategies teach this. Work on your tasks in priority order until they're finished, and then and only then move on to the next item on the list. There is a significant amount of sense to this approach. Most of your project constituents know they will get another chance to hit their lower priorities later, but this is their chance to get their top needs or problems addressed.

Avoid thinking that you can get away with delivering on the latter strategy—delivering quantity instead of quality—without compromising your current and future projects. Even if your business owner thinks that quantity over quality is acceptable, you always have someone say, “The project accomplished a lot, but they missed out on the top opportunity.” More than likely, that person is an executive, project sponsor, or someone who can influence the perception of the project and of you as a project manager. Delivering on bottom-up thinking can be attractive, but the people keeping score are unlikely to agree with your decision.

The final key to working top-down is to think about how you want the project to be remembered. The day or week or month after launch, everyone will know what you delivered and will be able to recall everything that is new. But a quarter or a year later, all of that will fade into a distant memory, except the top thing you delivered. If that top thing isn't memorable, then your project won't be memorable either. You want people to remember your project for its key success: its “top 1” contribution. Otherwise, your project is just like all the others.

Outside-In Requirements Gathering

Now that you understand about focusing on top-down requirements gathering, let's examine one more directional topic. Just as you want to make sure you're delivering on the highest-value requests, you also need to make sure your requirements are viewed as outcomes rather than as system or internal requirements. There is plenty of literature you can read about the specifics and techniques that make up outside-in requirements gathering, but

¹ Another “skill” of program managers is oversimplification.

the concept is the same as in the other sections: deliver what your clients want and need.

For instance, let's go back to a dark time in our history, the Y2K problem.² I worked on one of these problems myself. As a team lead at the time, I was incredibly focused on the idea that we needed to move from two-digit years to four-digit years, to account for the change from 1999 to 2000. Looking back, I realize that not a single one of my customers was asking for this. They were telling me, "Make sure our systems don't break when the date rolls over." The requirement to move to a four-digit date came from within my development team. That is, the hard requirement to move to the full year was generated from inside my team and was pushed out to the customer.

Sometimes this isn't a bad approach. Your customers don't necessarily understand the systems and limitations they need to deal with. Your team may say something like, "In order to solve this problem, they need to get used to typing out the full year," or something equally pithy. These are real constraints and requirements, and they need to be discussed, communicated, and agreed on. Put another way, if your customers are going to object, better to find out now than after the solution is designed and delivered.

Note Be sure to understand the difference between a requirement and a constraint.

But it's important to note that these aren't requirements. These are *constraints*. In order to meet the requirements, these are the specific limitations the customer faces. Constraints are okay. In fact, they're a very good way to understand the cost of what customers are asking for and how much effort or inconvenience will go into getting your customer the solution they want. But in the days before Y2K, virtually no customer was asking for a four-digit year. They were asking for their systems not to break. Mixing up requirements and constraints this way can result in communication challenges.

This isn't to be confused with what many project-management guides call *nonfunctional requirements*. These are requirements about how the system needs to operate; they aren't constraints. Topics such as performance, operability, security, and maintainability are true requirements that your team needs to balance and prioritize along with all the functional requirements. Often, these types of requirements can be worded as if they're being

² If you weren't around at the time, look it up.

viewed from the outside: for example, “Must load within 8 seconds” or “Must be able to be maintained by an untrained operator.” Both of those are true requirements that you need to manage.

Too often, project teams view their requirements about solutions as actual requirements. When you’re in this stage of the project, remember the Y2K problem example. No one was asking for a four-digit year, but now you see it everywhere. Did we get it right? Was going to the full four-digit year the correct call? It may have been. But many of us treated it as if it were the stated requirement, when it almost never was. Now, every time you’re forced to type in **20** at the beginning of the year your credit card expires, you can thank those of us who made this a requirement.

The Value of Focus

Let’s talk briefly about the value of focus. Much as in the discussion of top-down and outside-in requirements gathering, there is hidden importance to focus. The ability to focus is one of the scarcer resources you have to manage. You can hire a lot of people and do a lot of things, but you, as a project manager, can only pay attention to a few things at any given time. Some studies have shown that you can keep only four things on your mind at once.³ What this means to you, your project, and your clients is that deciding which requirements get your focus is a nontrivial decision. Deciding which requirement is number four instead of number five or number six can make the difference between that requirement being successfully met or not met at all.

Note Good project managers can focus on about four different items. Some can handle less than that.

A good way to message this issue is to be completely transparent and say something like, “You just made that my fifth priority. I’m pretty good at counting on my fingers, but once I reach my thumb, I start to get confused...” Yes, that’s a joke, and isn’t to be taken literally, but it does get the point across: you’re focusing only on the top four priorities (maybe fewer), and although you can surely give the old college try to numbers five and up, you can’t and won’t promise anything.

³ I didn’t make this up: www.livescience.com/2493-mind-limit-4.html.

You want your customer, client, or sponsor to value your focus. In return, you want to respect what you're focusing on. If you're going to actively manage four separate threads, then make sure you complete all four successfully, even at the expense of numbers five through infinity. If those lower priorities are actually important, then you can always propose a follow-on project, or some other effort, to pick them up. No project manager gets a pat on the back for partially delivering ten different requirements instead of locking down and solidly launching the top four. This isn't *Family Feud*—this is pure value-focused management.

Communication Strategy

Determining project requirements is key to success of a project. However, equally important is your ability to communicate your decisions to the larger project community. A very large, very unreadable document may hold up in a court of law or be useful if you're working on a defense contract. But most of the time, attempting to cover yourself with large documents doesn't do you any favors. Although these types of documents may let the project proceed and eventually deliver, you're unlikely to be called back for another project if you don't deliver what the customer truly wants.

This is where top-down, outside-in strategies can help you. Most people want to know whether their requirement is in scope or if it isn't going to get done. Being able to describe to this community what functionality your project will allow and what it won't goes very long way toward communicating your requirements decisions. This is also why limiting the number of primary use cases or user stories is essential. Focus your project community on the half dozen or so new things you can accomplish, not the 20 or 30 items that go along with them or the 200 or 300 things you can't do as part of the project.

You want everyone to think of your project and its top few bullets all at the same time. For example, when people think about Project Rodeo,⁴ they should immediately be able to list the very few things the project will deliver. You read previously about the ability to keep four things in mind at any one time. Although that was in terms of the project manager and project team, it's also true of your customers. Any bulleted list of features beyond four become noise. It doesn't matter if the list is 6 or 60 items long. You need to continually focus your customer base on the top four things you're delivering and see that they remember them.

⁴ I'm not embarrassed to say that this was the code name of one my favorite projects.

How do you do this? Figure out a way to describe your top 4 user stories in 25 words or less each. At the top of every project document, update, e-mail, and other communication, list them. Be open and honest, and say, “This project will do the following things...” This messaging should exist on your project portal and even in your e-mail signature if necessary. This 100-words-or-less message becomes the so-called *elevator pitch* for the project and the anchor people can use when they decide whether your project should continue.

If your insistence on having these 100 words at the top of every presentation and e-mail become a running joke among your team and your customers, don’t take offense. To the contrary, you should take pride in the fact that everyone involved has read those points so many times that they feel comfortable making fun of you for continuing to include them. They’re that important.

Note You can say 100 words in less than two minutes. It shouldn’t take you much longer than that to hit the highlights of your project.

After a few dozen or a few hundred times, your team and your customers should be able to spout most of this information from memory. You should continue to stress it. At this point in the project, you’re no longer trying to sell the project to the project team—they have already bought into it. And you’re ready if an executive asks you what is going on, or you get cornered by a project sponsor who wants to know what features they can expect. But you aren’t ready for an executive who isn’t keeping up with the project to ask for an update from someone else who also isn’t keeping up with the project.

The worst thing that can happen in this case is for the person to say, honestly, that they don’t know, or that they don’t understand what the project is trying to do. Although these people probably haven’t seen your 100 words repeatedly for the past several months, they should have seen them a few times and be able to list one or two of the highlights. Failing that, they can go back to their desk, open last week’s e-mail, find your list of key highlights, and immediately forward them to the person who asked.

Sure, your project is doing many things, probably too many to list in a document that anyone would read. And most certainly your development team couldn’t proceed without fully understanding requirement 4.2.1.3(a) and its implications on the rest of the systems. But that’s not what I’m talking about.

I mean this simple question from an executive: “What am I going to be able to do after the project launches that I can’t do right now?” If you can’t answer this question, then you should be prepared to have your project scuttled or restarted. If your value proposition takes you more than a few bullets to describe, then your project probably isn’t worth doing in the first place.

Sign-off

The milestone of “Requirements sign-off” is an important one. At this stage, the project team and the client sponsor or client indicate that they agree with what the project will be delivering. This is often the result of weeks or months of work, analysis and negotiation. The two kinds of agreement that we look at next are formal sign-off, the actual language that creates a contract of sorts between the client and project, and informal sign-off. Informal signoff occurs via conversation or through a relationship, where you ask the question, “Is this really what you need?”

Formal Sign-off

Virtually all project management guides talk about the importance of formal signoff of project requirements. Some organizations still do a literal signoff, where the project manager has a clipboard and a pen, and stakeholders physically put ink to paper in order to show acceptance. This might strike people as something between silly and archaic, but there is a lot of symbolic value in asking people to sign their names. Before people are willing to put their names down for posterity, they generally make sure they agree with whatever it is they’re signing.

If used this way—as a way to sink into people’s minds that this is really it—then a formal signoff can be a very powerful tool. You’re asking people to make a final determination about what is in scope and what isn’t. If used in any other way, particularly as a weapon for later, then it can still be a powerful tool, but one rife with stress and arguments. In some cases, such as high-end consulting engagements or projects where changing requirements is very expensive, this is exactly the kind of situation you want to create. If you’re running a normal project for your company, then this is the kind of environment you want to avoid.

Note A formal signoff can be a very powerful tool if used intentionally and carefully.

You want to view this signoff step as outlining baseline requirements. You did a great job gathering, consolidating, explaining, and clarifying what your project ought to do. You've distilled what you've been told and figured out over the course of weeks or months into a single document that describes everything you know about the project to date. You should be proud of this accomplishment. But the key phrase to remember is, *to date*. The document you just put together codifies everything known about the project at the time that the document was written. Attempting to insist that from this point on, nothing is allowed to change will drive you mad.

Not only will it drive you crazy, it will also drive your sponsors, team, and clients crazy—all for virtually no gain. Your requirements document can either remain open and living, with you updating it as you learn more or as things change, or it can be a point-in-time document that baselines what you thought the project was going to be at the time. Both are valid approaches.

If you work with your project stakeholders to help them understand that the purpose of the document isn't to lock out changes or to require significant work if something has to happen, then you can have a much smoother and probably more productive conversation about it. Unless you're actually signing a contract, don't treat it as one. Treat it as a recording of everyone's understanding at that specific point in time, and acknowledge that things will change, the market will shift, and better ideas will be formulated. But for now, right now, this is what everyone seems to agree on. Framing your signoff that way will greatly improve your interactions when you're standing in front of them with a notepad and Sharpie.

Informal Sign-off

The phrase *informal signoff* sounds a little like an oxymoron. Either they're signing off or they aren't, right? That would be true if every signature on the document carried the same importance to you. The *to you* in that sentence is important, as well, because there are people who need to approve your requirements who have high-level titles or command respect but aren't key stakeholders for your project.

For instance, years ago I ran a project for which our Vice President of Tax was adamant about how we needed to tax the sale of postage stamps. (We were actually selling stamps to customers with a small markup.) He wouldn't sign off on our requirements unless his tax system was in the document with assurances that we would build it to his specifications. We heard this and treated it as an important requirement; if we did something

wrong with regard to the tax code, we would probably get in trouble somewhere down the line.

But the VP of Tax wasn't my customer, a client, or even a key stakeholder for the project. My job as project manager wasn't to please him, only not to anger him or cause him to derail the whole effort. I did have a little bit of a vested interest; I'd probably need him on future projects, so I didn't want him thinking of me as the Guy Who Messed Up Our Taxes, but that was a minor consideration. What was important is that my key sponsors agreed with the document they signed off on.

Following up

Getting signoff is an event, and one to celebrate. It should mark a true milestone in the project that that took weeks or months of hard work, compromise, and tears to obtain. It's okay to take a quick breather and reflect on how the project has gone so far. Shortly after that, however, you should be asking the project owner, sponsor, or client if they like the document they just signed. This is an easy step to miss, but it's vital to the health of your project and team.

This takes the form of a simple question: "How do you feel about the requirements document?" It's rare that everyone loves everything on those pages. The document is the result of compromise, negotiation, giving in, and previously unknown rules and constraints. If you did a good job, you leave most people with the feeling that you did the best you could have done. As much as I love getting the feedback that I did as well as could be expected, I'd much rather hear that I did well, full stop.

Note Doing as well as can be expected isn't the same as doing well.

What you want to hear at this point from your sponsor is the *but*, as in, "I like what we've done, but..." This means either that there are things missing from the project that the client thinks are important, or that some negotiation didn't go the client's way. You need to find these things out now. It's important to determine whether the project as described is still worth doing, or if the place you wound up is no longer compelling enough to move forward. The worst thing you can do is to plow ahead with your project simply because the requirements are acceptable. You need to keep communicating

with your key stakeholders and make sure they like what the project will be and aren't just settling for what they wound up with.

Summary

Requirements gathering and transcribing is one of the first actions and deliverables of a project. A certain order will greatly improve your outcome, if done properly. Too often, project managers begin to document requirements from several different sources without first understanding who the customer of the project actually is. The very first thing you should do is figure that out. More to the point, until you figure this out, you don't truly have a project.

When you know who the project's ultimate customer is, whether it's an outside client, an executive, or your boss, you can ask, "What are we trying to do?" Focusing top-down and outside-in on requirements elicits the stories and use cases that you need in order to build your system. If you start talking font sizes or logo colors, then you've greatly missed the point, unless you work for a company that makes fonts or logos. You should be relentless about finding out who is benefitting and what they want or need. Once you figure this out, sorting out the details is a much easier task.

You can only focus on so many things, and this is something you should be ready to admit. Of course, you're willing to try to do more, but you can only promise the stuff at the top; so choose the things at the top carefully. From these top items, you should be able to craft your pitch for the project so that everyone involved and anyone who asks can get a two-minute summary of the highlights. Make sure this statement is compelling: if it isn't, start over with new requirements.

When you have all this down, you need to take two different paths with your synthesized requirements. First, you need to get the formal signoff, or approval, of everyone who could derail the project. These are the people who don't care if you succeed, as long as you don't mess them up in the process. This is painstaking, often frustrating work. But even after you have formal signoff, you aren't finished. Eventually you have a document of 2 to 2,000 pages, which contains the summary of all agreements and knowledge to date. This is the time to ask the client or sponsor, "Now that we know what this project is, do you still want it?"

That question may yield a wide array of answers, from an excited "Yes" to a bland "Not really." Usually the response is something in between, leaning more toward yes than no. But there is no way for you to manage your

client and stakeholders without knowing where on this spectrum they fall. No project manager should consider going ahead with a “not really” project. You’re wasting everyone’s time if you do.

Prioritize

“Which is more important: critical, must-have, or essential?”

It would be difficult to overstate the importance of prioritization—in your daily life, in projects, or even in deciding what to have for lunch. Without knowing what is most important, what comes after that, and what comes after that, it becomes very difficult for your team and you to decide what to work on next. A significant source of drama in projects is caused by miscommunication and misperception about priorities, and how they differ from person to person and project to project.

Even worse is when the priorities of the project were never decided in the first place, leaving a vacuum of prioritization where order and guidance should exist. It’s forgivable for people to have debates, arguments, and negotiations over the priorities of a project; in fact, it’s both desired and expected. Unforgivable, however, is not having such debates, and not having thoughtful deliberations about the decisions you eventually make.

If you were to let these discussions happen without direction, you would wind up with nothing but top priorities, and no real order from the chaos of goals and objectives. It isn’t unusual for customers, clients, and sponsors to view each and every requirement as an absolute must-have, worthy of holding up the entire project in order to get it completed. This is a lofty desire that might work in a perfect world with infinite resources, time, and budget.¹ Unfortunately, most projects don’t live in that world, and you’re tasked with making hard choices, or at least helping others make them.

¹ But probably still wouldn’t.

In Chapter Three, I talked about how a normal project manager can handle four different major threads of work at any given time. This means distinguishing between objective number four and objective number five—it's critical when deciding what to do. Without doing the hard work and making the hard decisions about what is important, you may know which things are of high importance and which are low, but you don't know which are high enough for you to pay close attention to, which aren't. And it's likely that too many things will wind up being labeled critical—probably more than your project could ever hope to accomplish.

This is an area where project managers differ from engineering managers and executives. Other disciplines sometimes talk about prioritization, using the idea of *above the line* (A-priority items at left in Figure 4-1) and *below the line* (C-priority items). Everything above the line is committed as in scope for the project—that is, it's part of the project—and everything below is out of scope. This approach is alluring and sometimes better than nothing, but it's not correct. Savvy project managers, and those who wish to limit drama within their project, have a second line. And this is the line you need to pay attention to. As you can see at right in Figure 4-1, being able to commit to a certain level of delivery while allowing yourself the flexibility to sacrifice important items (the B-priority ones) for truly critical items (the A priorities) can be important to the health of the project.

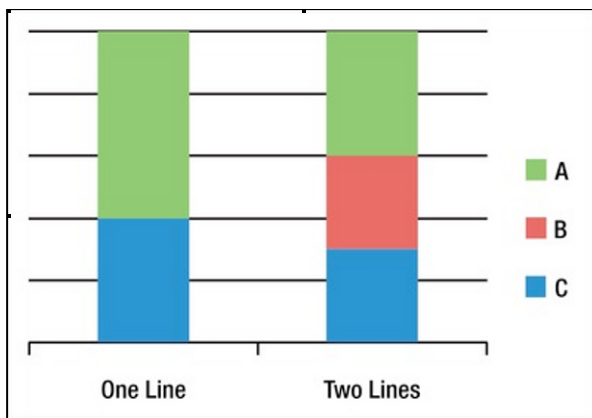


Figure 4-1. Two-level prioritization

■ **Note** Project management is often about drawing lines. One line is often not enough.

In your world of everything being critical, you have to think of the two different levels of must-have. It seems almost silly to consider them differently, but imagine your house. In New England, a reliable source of heat is vital in the winter for survival. Without it, you die. However, you don't need a four-zoned house with climate control in each room. In fact, you can survive without heat in every room. Of course, your house would be much better off if you did have it, but the entire family could sleep in the same room and not perish.

This is an example of the two kinds of critical requirements:

- Requirements without which the project will be a total failure (A items)
- Requirements without which the project will be less successful than it could have or should have been (B items)

Without a doubt, both are critical. You certainly don't want to fail, nor do you want to underachieve. Part of the reason you run projects is to launch successful things into the marketplace, right? But for many reasons, you need to know and understand this distinction.

The Importance of Prioritization

One of the key exercises that project and portfolio managers must go through is prioritization. Even the most rookie project manager can take a list of requirements and draw a line somewhere, asserting, "Stuff below this line is out; stuff above it is in." More experienced project managers draw that second line: maybe it isn't as dark, and maybe they don't draw it with permanent marker, but it's there all the same. This is the "must-have vs. should-have" list, or as I like to call them, As and Bs. If all you have is a list of things that must happen, then everything is top priority—and everything is bottom priority. In other words, your list is useless.

Why is having the second line so important? Why is a project nearly certain to fail without it? There are many reasons:

- You and your team should always be focused on the highest-priority items on the list. If you don't know which things are high priority and which are high/medium, then how do you know what to work on next?
- When things happen and you lose time, you as a project manager need to know what functionality can suffer in order to make the important stuff shine. To make this work, you need to know both sides:

which is the important stuff, and which part can suffer. And you need to decide this before you need it, and not when the crisis hits.

- Without it, you're setting an unrealistic expectation—that you're perfect. Unless you draw the line incredibly tight, then you're pledging 100% delivery. This is no way to properly set expectations with the client.
- Conversely, if you do set the line very conservatively, then you aren't being aggressive enough. You need the ability to over-deliver, or wow your client, in order to make up for misses somewhere else. Bs clearly aren't worth as much as As, but delivering on them without adding additional cost to the project does draw the client's attention.
- You need to know how to staff your team. Because you should be optimizing for the A requirements, you know to put your better, more experienced people on them. Having a pile of B requirements does a lot for you. It allows you to build bench depth and strength; it can be used for employee morale if a B is particularly interesting; or it can be used as a bargaining chip with QA, your team, or your client.

I've seen requirements lists 40 items long, with 39 of them must-haves. Such a list may as well not exist. As a project manager, you need to relentlessly put the work in some kind of priority order. That may mean literal cardinality; but really, just figuring out what is a high priority and what isn't makes a gigantic difference in your ability to deliver.

Some things to keep in mind:

- No B requirement should ever cause an A to slip. You won't get any love from your client if you miss an A because you were delivering a B.
- Bs should never be critical to the project, business, or client. You're sending the message "The team will do as many Bs as possible, but maybe not this one." If the client can't stomach that, then you're looking at an A (or a larger issue).
- There is no shame in working on a B. Bs exist for lots of reasons including training, scheduling, keeping a team together, and so on. Just because an item is lower on the list doesn't mean it isn't hard or interesting. Just know that if you're working on one, you may be disrupted at any time to work on an A priority instead.

Without prioritizing your "in" list, you wind up in an impossible situation. You've sent the message to the client that they can expect perfect delivery,

and you've removed any flexibility that you may have had when running the project. Don't do that to yourself.

The True Critical Path

Project managers generally know what the critical path is for a project, how to determine it, and how to represent it. Traditional project-management techniques tell you to find the dependency chain that takes the longest to complete; when that sequence of tasks is complete, your project is finished. This is very good advice, and I would be foolish to try to convince you otherwise; the end date of your project is indeed when the last critical task is complete. I'd rather redefine what *complete* means in a no-drama context.

The traditional definition of *complete* is also the dictionary definition. The project is complete when all the tasks are finished. Any task that lies on the path that takes the longest is therefore on the critical path. What is hidden in this is that it uses the wrong meaning of *critical*. This strategy uses *critical* in the medical sense: if this part of the project is running late, then your project is in critical condition. What you ought to do is use the definition of *critical* that is closer to "important," indicating what you're really trying to accomplish.

■ **Note** You should use the definition of *critical* that means "important" rather than the one that means "unhealthy."

Somewhere in your project plan lies the delivery that your client, sponsor, or customer truly cares about. Depending on the project, the client, and the need, this delivery can be as simple as a beta version of the product or as complex as needing the client's entire staff trained and using the new system. This milestone can happen six months before the project is over or six years later. As project manager, you must identify what this critical delivery is and where it falls on the project plan.

Many clients believe in the concept of *sine qua non*, which means something like "that without which": no matter what else happens in the project, if this piece or function isn't a success, then neither is the entire project. This definition changes from project to project and client to client. However, it's your task as project manager to figure out what this critical deliverable is, and when and where it's going to be delivered. Then, during your project,

you need to keep track of not only the traditional definition of the critical path, but also the no-drama definition.

If you have a project with multiple stakeholders (and don't we all), you may have many of these critical deliverables. For instance, if you're creating a new system to replace an existing one, one stakeholder may be focused on when it's in place, whereas another may be mostly interested in when the old system is completely deleted. This is not only okay, it's the information that you need: you need to know when each of your clients or stakeholders will be satisfied with your delivery. And it's expected that none of these critical deliveries line up with the "end" of the project. Being finished with the work doesn't necessarily mean the value has been captured. This is an important lesson to remember.

Prioritization Games

Even with this in mind, it's easy to fall into the trap of playing games with your priorities. It's a common ploy to put the B+ requirements (those that really are should-haves) early in a project schedule. The project owner then saves the actual A requirements for later, when you have no choice but to attempt to deliver them. They get not only all their A requirements, but also many of the Bs, simply because they convinced you to do the B requirements earlier in the schedule. Don't allow this.

This strategy usually manifests as a client, sponsors or customer being very agitated or animated about an idea, a concept, or a requirement that sounds important but isn't truly critical for the success of the project. The argument supporting it seems logical, and even an objective observer would agree that whatever it is, it's worth doing. But there is a huge difference between something that is worth doing and something that is vital to the success of a project.

One of my project managers once ran a "Ratings and Reviews" project, allowing customers to rate and review our products on the web site. For some reason, many of the executives were worried about abuse and the use of foul language or other offensive content in the reviews. They were so worried about it that they wanted us to build in an automated bad-word blocker so that if someone typed in a word on a predetermined list of known bad words, the submission would be rejected. This sounded like a valid, very reasonable request.

However, we were already building in a mechanism for a human being to approve comments before they went live to the public, as well as a way for

this approver to delete comments that had already gone live. So, for an inappropriate comment to be exposed to customers and stay exposed to customers would require a human being to approve it in the first place and also neglect to remove it later in the process. In other words, working hard to automate rejecting the comment wasn't all that valuable.

Additionally, my project manager asked a reasonable question: "How many inappropriate comments do we expect to get?" The answer, depending on who we asked, turned out to either be a really low number or a much more truthful, "We don't know." With this in mind, the project manager was able to call this a non-requirement. He pledged that we would monitor the comments that came in, and if abusive ones were coming in with a significant regularity, then we would absolutely build something to automatically filter them out. In the past 2 years, we've gotten about one really bad submission every 6 months, which takes a reviewer about 30 seconds to delete. Had we built this functionality, we would have saved ourselves about a minute of effort per year.

■ **Note** Beware of requests that sound critical but actually add little value.

At one point, our sponsor declared this automated review requirement to be a showstopper for the project. I was willing to play chicken with him. In fact, I was willing to sign up people on my team to actively monitor comments, to ensure that nothing untoward was released for public consumption. This allowed the project team to focus on the true A requirements and deliver a solution that worked and provided the value that we needed the project to handle. Had they spent any energy on this B requirement (which turned out to be a C), then something at the top would have suffered, and actual value would have been lost.

Pairwise Prioritization

Prioritization can be difficult in the best scenario. If you're launching a product with ten similar features, deciding on the prioritized order of those ten can be a chore both intellectually and emotionally. But once those ten features become disparate or not comparable to each other, the exercise becomes nearly impossible. How do you compare the safety of your product versus the number of colors it's available in? How do you choose stability

over speed, or transparency over correctness? Although these aren't items that are mutually exclusive, they're hard to compare against each other.

For instance, years ago I ran a project that included a vendor selection for a critical piece of functionality. The list of requirements included "Accuracy," "Performance," and "Is produced by a company that won't go out of business right after we buy it." I understood that all these requirements were important, but it was difficult to decide how to balance them. We had an option that would have been tolerable; it was from a company that was 50 years old and incredibly stable. However, none of the evaluation team thought it was the best option when considering the other requirements.

We had a total of five top-level requirements for this search. The other two were "Stability" and "Ease of use." All of these five are laudable, good requirements. But trying to find a solution that maximized all five would lead us either to a bad answer or to no answer, because if we tried to achieve all five requirements with equal weight simultaneously, it would take us years to find the right solution.

What we did was put the five top-level requirements in order, using what is sometimes called *pairwise prioritization*, which Wikipedia calls the *Analytic Hierarchy Process*.² This process requires taking all your requirements, constraints, and needs, and comparing them against each other in a pairwise fashion. The needs of a project are usually very difficult to prioritize in a straight line, because they're so different from each other. That makes it difficult to determine which is the most important and which is the least. Even though it is difficult, pairwise prioritization can be extremely illustrative. You will be asking people to compare things such as security vs. performance, or usability vs. advanced functionality, things that often aren't compared to each other. Having this conversation not only helps to clarify requirements, but it can also give you insight into the overall needs of the client.

When I learned this method years ago, the sample problem was, "Where should we go for lunch?" We gathered the following requirements:

- Should be close to the office
- Should not be Chinese food
- Should allow us to be back within 90 minutes
- Should cost less than \$20 each
- Should have table service

² http://en.wikipedia.org/wiki/Analytic_Hierarchy_Process.

This is a decent list, but it's very difficult to prioritize without doing a true head-to-head comparison. Because we all had a very important meeting in 90 minutes, that priority was more important than being close, or costing less than \$20, or offering table service, or the kind of food. So, it came first. We decided that we wanted table service (and the chance to talk with each other) more than we wanted to spend less than \$20. We then decided that spending \$20 or less was more important than it not being Chinese—only one person didn't like Chinese food, and either he wouldn't come, or he could just eat rice. And then we decided that being close to the office was least important, as long as we were back within 90 minutes. The final prioritization was as follows:

1. Should allow us to be back within 90 minutes
2. Should have table service
3. Should cost less than \$20 each
4. Should not be Chinese food
5. Should be close to the office

As we went down the list, each thing on the list was less important than the things above it and more important than the things below. We shouldn't seek to spend less than \$20, for example, if it meant we couldn't talk with each other. But even if we had a great solution for items 2–5, if we couldn't get back within 90 minutes, then we couldn't go. This pairwise prioritization—"This is more important than that"—is a very clear, but very challenging, way to prioritize. But the outcome is well worth the effort.

Back to the business problem. Using this method, the final ordering was Accuracy, Speed, Stability, Company Life and Ease of Use, in that order. This prioritization had a profound impact on how we were talking with potential vendors. No longer were we focused on how easy their system was to use, or whether we thought they'd remain in business. Instead, we were able to spend 90% of our time on the accuracy and speed of their systems.

Due to the pairwise prioritization of requirements, we came up with a much different answer than if we had tried to value all the requirements equally. We decided to choose the vendor that had the most accurate results, with the fastest speed, and we sacrificed all the other needs unless they were huge detractors. If the first two requirements were super-positive and the bottom three were neutral, then that was enough to put a vendor ahead of a solution that was merely positive on all five. As it was, we chose the solution that met the accuracy and speed needs, and the vendor promptly went

out of business (actually, it was bought by another holding company) within a year. That system is still running, and we don't regret the choice.

■ **Note** Pairwise prioritization can tell you that priorities 1 and 2 may be more important than 3 through 10 combined.

Pairwise prioritization can also be powerful when dealing with project requirements. You need to deal with limited resources, budget, and time. You simply can't spend an equal amount of time on priority 10 and priority 2. If you do, then you endanger delivery of all your priorities. This rank ordering should happen very early in the project, when everything seems okay. For example, you can ask, "If making sure the product passes compliance means we can only produce it in one color, is that okay?" Seems like an obvious answer—it's no good to have a noncompliant product in ten colors if that means you can't sell it. But the power of this question and this decision shouldn't be minimized. Focusing on the important things is what project management is all about.

The Value of Deprioritization

Let's talk for a moment about the flipside of prioritization, which is active *deprioritization*. This refers to the explicit decision *not* to do something. When you have your full list of requirements, you wind up with things that fall in three buckets: those you must do, those you should do, and those you may do if you have time. That last category is code for "don't do." In fact, if your program manager finds that you're working on the "may do" tasks in your project without a very good reason, he is very likely to cut your budget or resourcing.

In Figure 4-1, the "may do if you have time" items are the C requirements. They're out of the project, won't be delivered, and should be thought of as if they never existed. The message to your client, sponsor, or customer is simple: "We aren't working on this." Why is this valuable? It's an interesting and fair question. Over the course of a project, you're barraged with requests, questions, and invitations to discuss lots of things. Among those things are items you've already decided are C requirements, or items that are specifically and intentionally out of scope for the project. Unless you have a strategy for handling requests for analysis on low-priority tasks you'll

spend a lot of time—probably including a lot of your own personal time—managing things that should never have risen to the top of the list.

Agreeing ahead of time what is out of the project is a valuable time-saver for you and your team. The reason may not be obvious at first, but consider this situation that happens often: an idea comes up that is tangential but somewhat related to your project. Although it seems like a good idea, and perhaps it's a requirement that should be done at some point, it has no bearing on the success of the project you're currently managing. However, the allure of the idea's quality compels you to discuss it. Or worse, you fall prey to the notion that you need to discuss it so you can know for sure that you want it to be out of scope.

■ **Note** Don't be tricked into discussing something just to be sure you don't want to discuss it.

This kind of thinking wastes the valuable time, energy, and focus of your team; and as I've said before, focus is one of the more scarce resources you need to manage. Unless the C requirement somehow helps you reach project objectives in a way that they aren't being met already, then investing an hour thinking about it takes an hour away from working on the things that are making the project successful.

Project managers often fall into this trap, saying things like, "It's only one meeting," or "I won't make any commitments." Unfortunately, both those statements tend to turn out to be false. Yes, it's only one meeting for now; but when you've given the idea some of your attention, then the next time someone wants to talk about it—next week, next month, or next year—they will turn to you. Even if you successfully deflect the idea to someone else, you still need to help transfer whatever knowledge you have to them. And of course you won't commit to adding a requirement to your project—you're more competent than that—but you're committing a portion of your brain space to it. And even that is too much.

How to Deprioritize

The proper way to deprioritize is both simple and difficult. Often, clients are unable to understand the difference between "no" and "not right now." But that truly is the message you need to deliver. Their idea might be a good one. In fact, it may be so good that it's worth stopping the project entirely,

to focus on that instead. Even if that is the case, it's almost never the correct course of action to add this requirement to your current effort.

Chapter 12 discusses success criteria for a project and the fact that no project should begin without that list. You need it so you have a good way to help prioritize your requirements. When something comes up that sounds like a good idea but that doesn't get you further along your project's success path, you need to say so. And this is where the difficult part comes in.

I once ran a project for a company that provided data to salespeople. It was a subscription-based project, and the main goal was to reduce *churn*: the number of people who were cancelling their subscriptions. Increasing the number of months that the average subscription lasted by three would have had a very meaningful impact on the company's bottom line. In the course of discussing ideas for how we could do that, the idea of adding a premium offering into the package was brought up.

This was a pretty good idea. We could add a premium data source as an option to the product. But to do so, we would have to charge more. So, as project manager, I was forced to ask the question, "Will adding this option decrease churn?" The answer was no, it wouldn't—but it would increase the company's profit from those who took the option. This was a very helpful discussion, because I was able to ask the business owner if the most important goal was decreasing cancellations or increasing the value of the subscriptions that remained. I was unwilling to accept both as an answer, and the business owner knew and understood this. She chose decreasing cancellations, and the premium option was put aside for the length of the project, not to be considered again.

■ **Note** A good idea that doesn't further the goals of your current project sounds like a great idea to work on—in another project.

This is precisely the level of attention you need to pay to your prioritization activities: how does this idea, requirement, or task help further your goal? If the answer is "It doesn't, but it's a good idea," then you're in a very good position. You can agree with the person who made the suggestion, and say you also think it's a good idea and one that should be followed up—just not now, and not as part of this project. Short of changing the success criteria for whatever you're currently working on, anything that doesn't help you reach that goal is keeping you from it.

Summary

Prioritization is vital to a well-run project. Without knowing what the key deliverables are, you wind up working on many things without any real sense of their importance. This can cause you to thrash about from task to task in an uncoordinated way, making lots of movement, but possibly not making much progress.

Good prioritization tells you the order in which to work on tasks and also reveals where the true value in the project lies. This priority list may be different than the tasks that take the longest to complete. This concept of a critical path is different than the traditional project-management definition: it's the path toward delivering the critical part of the project, rather than the path that is most likely to make the project unhealthy.

One trick to get to this kind of prioritization is pairwise, or hierarchical, prioritization. Rather than deciding what the top ten requirements are, putting them in rank order may illuminate something you would otherwise miss. In many cases, the first three requirements are worth more than the rest of the project combined. Without doing this exercise, you can't know which requirements are at the very top of the value pyramid and which are only near the top. And not having this information can make running a project extremely difficult.

Finally, knowing when and how to deprioritize requests is also important. *Deprioritize* is another way to say "don't work on this." You need to have a firm grasp of your project's success criteria and how each of the things you're delivering meet that criteria. If you find that you're spending any effort on something that doesn't add to the value of the project, you need a way to stop. Even if the idea is a good one, that doesn't mean it needs to be part of your current project.

Without good and specific priorities, your team won't know what to do next and where to spend their time. Your client will be confused and frustrated by the progress your team is making; and although your project may still be a success, it will spend resources unnecessarily or even wastefully in pursuit of work that isn't key to the project. No-drama project managers do not want to find themselves in any of these situations.

Manage Change

“Sure, this changes things, but nothing has really changed...”

As Winston Churchill famously said during World War II, “He who fails to plan, plans to fail.”¹ In a similar way, project managers who fail to manage change are allowing change to manage them.

■ **Note** Change management is a part of project management, not the other way around.

Project managers are faced with change throughout the life of a project, leading many to believe that project management and change management are the same thing. You may even hear that planning isn’t necessary, because everything will change anyway. Any experienced project manager will tell you that this isn’t the case, no matter what it looks like from the outside.

But first, let’s talk about the two totally different kinds of change management: one that is within the purview of the project manager, and one that isn’t. When you read management books about effecting change or leading through change, what they’re often referring to isn’t how you run your project, but the *result* of your project. Whatever you deliver will have an impact on someone’s life, job, or day-to-day activities. Driving this change from

¹ I’m pretty sure it was Churchill, anyway.

concept through adoption is an art form that takes a large number of skills that may seem similar to those utilized by a project manager but are actually distinct to change management. Although you, as a project-management professional, should participate in, support, and influence this process, managing this kind of change is often outside the scope of your skills and responsibilities.

I'm talking about what I'll refer to as *project change*. This is the difference between what was originally agreed on as the full scope of a project, and what was actually delivered. Normally, this nets out to a list of things that were originally going to be in the project that was scoped out, coupled with a list of things that were originally out and are now in. Of course, it can also include straight adds or straight deletions; but normally clients want the total cost of a project to be somewhat close to the original, causing them to stay within shouting distance of the originally agreed-on scope.

A few years ago, I did an analysis of my portfolio and found that about half of the original agreed-on project requirements were delivered. This fact drove my team a little nuts. It suggested that all the work they did to gather, document, cost, and estimate requirements was wasted. A requirement being in the project at the beginning didn't mean it would be in the project at the end. This discovery was coupled with the knowledge that the average project added a little over 60% more scope than initially approved.

The net result was that the normal project in my portfolio cost about 15% more than initially specified and delivered less than 55% of the initial functionality. The team wondered, why should we bother making plans in the first place? I was thrilled with this question. Even though my team felt a little dejected, the answer helped show the value of what we were doing and how we were having a positive impact on the client, even in the things we weren't doing.

We took away two important things from the analysis, even if my team didn't pick up on them right away. First, the final cost turned out to stick remarkably close to the original estimate. On average, our projects were running about 15% bigger, but no project ran more than 30% bigger. This alone was a great outcome in my opinion. Even though we misjudged what needed to be in the project by roughly half, we were close enough on the overall budget that we could consider it a rounding error. A few people criticized this outcome, saying we "backed in" to the budget; because we had already baselined a cost, we made sure our final estimate was very well aligned with the initial figure. Although their criticisms were true, I had a hard time considering them negatives. Yes, we set a budget ahead of time, and we did a passable job of staying within it. How is that a negative?

The second important part of this discovery involves a theme that runs throughout this book: making decisions in a transparent way. Before a project begins, you spend a fair amount of time gathering and estimating requirements. And yes, it's unfortunate that nearly half of them don't move forward. However, a worse outcome would be to deliver on requirements that weren't going to provide the value the project needed! By fully understanding and specifying what the project wanted from us, we were able to consciously reject or remove requirements as it went along. By intentionally removing things from the project and adding other items of greater value, we were able to deliver a much better project containing much less useless functionality. If we hadn't done this work up front, we never would have known enough to make those decisions.

This process doesn't work if you leave it to chance. In order to properly manage project change, you must understand its importance and how it interacts with project methodologies. You also need to understand what causes project change, how it can impact the people on the project as well as the project itself, and how to best make sure that through change, you remain aligned with project goals. Not all good ideas should be pursued if you want to hit your goal.

The Importance of Change Management

Some of the primary causes of drama in projects include confusion and lack of communication. When people don't know for certain what is going on, or don't feel they're being communicated with, they tend to assume the worst. They then start seeking out their own information, causing friction and chatter. That leads to stress and anxiety around your project.

Change is by far the biggest source of stress and confusion in a project. Consider the project lifecycle. You spend significant time up front documenting requirements and creating a volume of paperwork for signoff that is circulated to everyone to read and approve. Then consider how much time is spent on changes, even large ones, and how they're communicated. Your project quickly goes from being one that is well understood by everyone, to one that is completely understood by relatively few people.

■ **Note** You generally spend much more time on initial requirements than you do on changes, even if changes represent more than half of the project.

Often, changes to a project are only communicated to those who need to know or are directly affected. This can include the team members who need to change what they're working on, the owner or sponsor who may need to know what how the budget or delivery will change, and anyone else who is directly impacted. But many other people are either indirectly impacted or are merely interested. These people used to know, more or less, everything that was in scope for the project. Now they don't know, but they don't know that they don't know.

This is usually okay because people don't really need to know things that don't affect them. In fact, communicating too much change too often may cause your project to look like it's disorganized or poorly managed. But picture this scenario: a fairly senior person on your project team is eating lunch with one of his friends at work, who isn't on the team. The friend says, "Hey, I hear you guys are only coming out with one size of the product instead of three, and even so, you're going to be six months late." Your team member, who knows about the project schedule, says, "Yeah, it's going slower than we hoped, which caused the delay. But we're still coming out with all three sizes." He was wrong; you did decide to reduce the scope to one size. You just made your senior team member look somewhat foolish; he knew about the schedule change because it affected him, but he didn't know about the change in the number of sizes available because it didn't affect him. Even though the reason you're in this position is perfectly logical and reasonable, you still wound up doing the wrong thing.

Change can be difficult to manage, and it's even more difficult to find the right level of communication required. It's worth recognizing that your project may be made up more of changes than initial requests, which can leave portions of your project team in the dark. It's worthwhile to understand what causes change, how it impacts different parts of your project, and what you ought to do about it.

Drivers of Change

Projects change for a huge number of reasons. Some drivers of project change are positive, such as capitalizing on new advances or new markets that were unknown at the start of the project. Others may not be as positive, such as problems in execution or lack of clarity in goals. Most of them lie in the middle, though, and are a result of a shifting landscape and evolving priorities. These can be both good and bad at the same time.

Understanding what causes change is just as important as understanding what is changing. You, as project manager, need to know if you're facing a positive, reactive, negative, or neutral change, so you know how to respond and how to communicate the need for the change to the team. As with many things in project management, knowing the reasons is sometimes more important than knowing the actual requests. Let's look at a few of the main drivers of change and what they mean.

The Business or Client Organization Has Changed

This means the actual organization structure has changed. Because projects are often approved months before they start and can run for several months, quarters, or years beyond that, it isn't unusual for the client or business organization to change shape. This can result from basic business restructuring, mergers or an acquisition, or a new CEO or business leader taking charge.

It isn't uncommon, in the face of organizational change, for projects to be re-evaluated to make sure they're in alignment with the new reality. Projects can be canceled outright; others change course, shape, or ultimate goals. If your project is canceled, it may not be a sign that the project wasn't a good idea or that you were running it poorly. It's a sign that the new organization values different things more highly than the previous organization did. If you can, leave your project in good shape, even lobbying for a little time to do so if necessary. Although it may be you who picks it up if it ever gets restarted, chances are it won't be. You not only want to leave the next person with a project in good shape, you also want to be known a project manager who knows how to close down a project. But be prepared to put your project away and move onto the next one.

■ **Note** Project change due to organizational change is so common that you might anticipate it before it happens.

If you're on a project that is facing material change to be better aligned with a new organization, make sure you're making this change thoughtfully and intentionally. Depending on how much things have changed, you may be faced with running a project that no longer makes sense. It may have half old requirements and half new ones, and they don't add up to a complete, cohesive

project. Beware of running two or three sets of project objectives mashed together. Running a project that has confusing or conflicting goals will certainly add drama to your project and should be avoided if at all possible.

New Advances Make Your Project Obsolete

The technology landscape is always changing. While you're trying to make something out of existing components, the companies that make those components are improving them, modifying them, and releasing new versions. Sometimes, this means the project you're in the middle of building has already been built (perhaps better) by someone else. This is usually a good thing in the long run, no matter how disappointing it feels when the change happens.

A long time ago, I was working for a web consultant shop. We realized that many of our clients were having the same problem. The web was stateless (it still is, actually), and the only way to keep track of customers was a technological nightmare. So, we decided we should be building a state solution, which we could then resell to our customers as part of the value-add equation in the engagement. Apparently this was an obvious, industry-wide problem, because by the time we were half done, several well-known companies, from Microsoft to F5, had come out with their own solutions. We were pleased as a team to know that we were on the right track and that the solutions other companies were releasing turned out to be very close to our design. But it still meant someone else beat us to the punch, and our project was now obsolete.

This kind of change tends to take one of two paths. The first is that the project is halted immediately while you analyze the available third-party solutions. Rather than constructing something, the project is now about evaluating options. The team that does this kind of work is usually different, and much smaller, than the team that was tasked with building a new solution. This changes the project into something else completely, and it's often better to have a different team do the work.

Sometimes, though, it moves from being a construction project to being one of integration. Your project plan should include how to get your new capability to work with existing systems. You probably built in the beginnings of a rollout plan, including some systems analysis, timing, and other factors. Much of this information and planning may still be valid. What has changed in your plan is *which* solution you're integrating and rolling out, although you still need to meet the same needs, goals, and objectives.

Depending on what shape this change takes, you may either pack up your stuff and move on or hunker down and figure out how to make the new

technological advance work. This should usually be considered a positive change; if so many people were experiencing the problem that it caused a vendor to create a commercial solution, then you were probably on the right track. Some of your team members may be disappointed that they didn't get to build the solution themselves, but at least you can still hit your objectives—and solve your client's problem.

Changing Priorities of Your Client or Organization

This reason for change can be one of the more frustrating for you and your team. Your project is still good, the need still exists, and your goals and objectives are still valid. However, the company has decided that it either has higher priorities or has more urgent needs, so it's going to drastically change your project, if not cancel it entirely. In my experience, cancelling or "putting on hold indefinitely" is much more merciful than the other option—a seemingly random change in direction.

This option, which I've seen far too often, is a challenge to explain to your team. Some organizations call this "repurposing" or even "redeployment." But what it means to you and your team is a lot closer to "do something else with the team that they aren't necessarily well suited to do." Your organization or client may have decided to lower the priority on its social-networking efforts, for example, to focus on lean manufacturing. Some wise executive says, "Can you take the social media team and put them on manufacturing? They're all engineers, right?" Adding to the joy of this kind of decision is that the project manager is rarely consulted before the decision is made and is even more rarely heeded. "There is no doubt that this is a well functioning team with a capable project manager. So why not? Your team is better than no team at all, right?"

■ **Note** From a high enough perch, all project teams look the same.

Worse than being cancelled, and maybe even worse than getting redeployed, is being allowed to continue. You may find yourself running a project that is still a good idea and always was. But the project is no longer a priority of anyone in the business or client organization. If you succeed, great. If you fail, no big deal. But you get a strong, implied message: "Please

don't ask for more resources or budget, because your project isn't that important." And good luck getting any executive or sponsor to pay any attention to the project. There is little to no incentive to complete it on time or to complete it at all. If you find yourself in this situation, you should be prepared to be redeployed sometime in the future, but do the best you can in the meantime.

New Sales Channels

New channels are a great example of how new shiny things can cause a disruptive change. Many executives believe that opening new channels somehow cures the shortcomings or defects in the current product line. They seem to be thinking, "Sure, here's a bunch of people who have no experience with our products, so they don't yet know what our weaknesses are. We should capitalize on that and make as much money as we can until they find out." This is a business strategy that I've seen used, and seen work.

Some fairly nonsensical requirement requests come from this kind of change. For instance, "We have bought air time on a bunch of children's shows on Nick Jr. Do you think you can make our product talk?" When you reply, "Uh... we make flux capacitors," you realize that has no meaning to the person with whom you're speaking. And you think about something pointy to poke them with.

New channels, in themselves, do nothing. Great: now you're on the shelves at Walmart and being sold by the Home Shopping Network. This doesn't change the fact that your product catches fire if left on for more than ten hours, right? But you get requests via the new channel to make changes that have no real positive impact, such as slimming the width of the packaging by half an inch to take up less shelf space, or to put a bunch of Samurai swords on the cover to appeal to the Asian market.

When this happens, you have a conundrum. Your clients, sponsors, and executives are probably putting pressure on you to deliver something that will have, at best, value for the next few months, and no value after that. This can even come at the expense of things that you think will generate long-term value to the company or organization. What is a project manager to do?

As much as it goes against your instincts, the best approach is to try to change the principles of the project. If your client and sponsor want you to optimize your time for whatever new channel they have created, be open and honest about it. Put "appeal to Asian consumers" right alongside "don't catch fire" in the list of things you're trying to accomplish. This will do one

of two things. It may shame your client into realizing that what they're asking you to do is the wrong thing, and maybe they will change their outlook. Or if that doesn't happen, you have permission to act as if this new channel requirement is as important as all the others. And getting this decision made explicitly will serve you well in the months to come.

New Legislation

This is an example of a change about which I have no good, clear advice. Sometimes, mid-project, new laws are passed, or new regulations are issued, or new compliance rules need to be adhered to. They come out of nowhere. Legislation, at its heart, is a constraint and not a requirement. So, you need to consider it a new project constraint that you didn't know about before.

■ **Note** Legislation is really a constraint, not a requirement.

I recently consulted for a company that wanted to participate in the Groupon phenomenon. The company wanted to use Groupon as a way to attract new customers and to try out offers in a captive, opted-in market. We came up with a pretty compelling offering, wrote all the backend web-site stuff to make it work, and released the offer into several test markets to see how it would do. In our initial testing, it did better than we imagined. Not only were more people buying our Groupons, but they were also coming to our site and buying other items that weren't on sale as part of the deal. This effort was incredibly positive.

But while this was happening, certain states were changing their tax laws with respect to online commerce. New York started to tax online sales to its residents. Connecticut, Illinois, and South Carolina quickly followed. We did the analysis and determined that although losing these four states hurt our business, it wasn't a killer; we decided to simply block sales in those states and continue. But when California enacted similar legislation, that left us unable to sell in 20% of the country. This was a problem.

In order to be compliant with state tax regulations, we had to make a significant change in how we calculated sales tax. Because no one from the tax team was on the project, and no one on the team wanted to touch tax, this caused a lot of drama. I could have non-tax people, who didn't want to do the work, make the changes in our tax systems. Or, I could acknowledge

that this change in requirements meant we had to acquire different skills than we had. I was faced with a bad choice that I didn't relish making.

In the end, I wound up splitting my project in two. We moved ahead with one team building out the capabilities we needed to make our Groupon involvement work, and we spun up a second team with a different project manager to work on the taxation part. Had we gone a different direction, I would have had two teams of engineers who were both unhappy with the situation.

The Impact of Other Projects

While you're running your project, many other teams are executing on theirs. Theirs could be in the same portfolio as yours, or they could be in a separate business unit. It may seem to all involved that their projects are unrelated to what you're trying to accomplish. Yet they can affect you, both negatively and positively, and often in ways you would never expect.

Depending on the other projects and how they manifest themselves in your world, you may be facing a new constraint, a new requirement, or both. For instance, a new corporate security project may require that all password entry fields need to be done over an encrypted line. This is a constraint, and you must add the functionality to your project. Sometimes, this information comes to you late—maybe even after you've launched—which causes significantly more work than if you had known about it in the first place. Other projects may turn a constraint into a requirement, such as a project that expands the number of characters available for a SKU from three to four.

■ **Note** If your company or client is running more than one project at a time, be ready for that other project to affect yours.

If your company or client is big enough to be running more than one project at a time, then there is a good chance that other projects will impact you in some way. It may make your life more difficult or easier, but it's likely to be a source of change to your project requirements. Finding this out before it's too late isn't the project manager's responsibility (although good project managers do have their sources of information). Paying attention to this kind of activity is the responsibility of your portfolio manager. But no matter how you find out—on your own, from your manager, or on release day—it doesn't matter: you will be impacted and need to be prepared.

Failure in Execution

This source of change probably causes the most drama. The failure can be something like a poor design, or a missing requirement, or unknown show-stopping constraints that are uncovered late in the project. It can also be something that isn't aligned with a single incident, but rather with multiple unfortunate decisions spread out over the length of the project. Whatever form it takes, there is little doubt as to the ultimate cause: somewhere along the line, the project team failed to execute.

I once ran a project for a data company that was supposed to take a huge amount of disparate information, create a hierarchical tree from it, remove duplicates, and display it in a graphical form to the user. The solution chosen by our engineering team was clever and produced perfect results. We were all excited by what we were going to deliver, and our tests consistently passed manual QA as being 100% accurate.

It turned out we were correct to an extent. The design was elegant and did produce perfect results—as long as there were fewer than 500 data points. Beyond that threshold, the report took so long to produce that it entirely consumed a server for several minutes—a big problem when we had thousands of concurrent users and only a handful of servers. We needed to completely redo the design.

This caused massive amounts of drama for the project. Not only did we have to go back and start over, but we also had to be doubly sure our design was performing. We had to scrap all the advanced features we were going to implement but still extend the project. In the end, we produced the bare minimum needed to call the project a success, but we went over schedule and over budget and didn't get to half of what we were supposed to deliver. Not one of our best moments.

These are only some of the drivers of project change. Some of them are understandable and can be good things, such as technological advances or changes that remove constraints. Some of them aren't as favorable, however, such as a client losing focus as they chase a new opportunity, or failure in the project team. Whatever the cause, change is inevitable, and it needs to be managed properly or it will soon manage you.

Be Aware of the Situation

If change has come to your project, it can be disruptive if you let it. It can add a lot more drama to the project than the actual change would imply. To

limit the amount of stress it causes, you need to be aware of the situation. This includes gaining an understanding of the change and its causes, understanding how the company or client has handled similar changes in the past, and determining how the change is likely to affect your team and other interested parties. This kind of situational awareness helps you determine how to best manage the change and limit the disruption.

The first thing you need to do is understand the change:

- Is it the result of a large shift in priorities, an external factor, or dissatisfaction with the project?
- Will it require a small adjustment or a wholesale alteration of the project?
- Will it affect the entire team, part of the team, or a single individual?
- When will this change take place? Will it happen immediately, or is it a few months away?

Having this information in hand helps determine who needs to know about the change immediately, and who doesn't need complete communication until everything is figured out.

Depending on the client or the organization, the first thing that comes out of people's mouths when they learn about the change may be, "Here we go again." If it is, this tells you a lot about how you need to manage your team and your project. Are changes like this frequent? Are they usually a disruption? Do they usually make things better, or are they generally viewed as random? Have projects in the past survived this kind of change? Have they thrived?

Then move on to the specific change. You want to find out three things:

- Whether your team members believe in the person who is requesting the change
- Whether they believe in the reason for the change
- Whether they believe in the change itself

You need to know if you're faced with drama not only as a result of the change, but also because of something specific to the change. It's not uncommon to have both to deal with.

Finally, you need to analyze who is impacted by the change, including people on your team and people in the client organization. Something that has a great impact on your business or client may have little impact on your project. The opposite is also true. And within your project, the change may affect the whole team, a small subset, or just one person (or maybe only you.)

■ **Note** People are impacted by change in different ways, and not often in the same magnitude as the organization.

Knowing the nature of the change, how the organization feels about change in general and specifically, and how much of your team or client will be impacted is critical to knowing what kind of drama you may need to manage. It can range from very little to soap-opera level. And of course, every project and situation can be different.

Be Aware of the People

As you determine who is affected, keep in mind that you need to consider two very different groups of people: the team or other people in your delivery organization, and those on the client or business side. And sometimes, you need to deal with both simultaneously. For instance, the organization deciding that now is the time to do a large-scale enterprise resource planning (ERP) rollout affects not only the finance folks in your business community but also the finance developers and IT staff who may be on your team.

So consider your team members and how they're impacted by the change that is occurring. This impact may be specifically what they're working on tomorrow or next week, or it may affect their careers or their positions in the company. Although doing anything about the latter is usually outside a project manager's scope, you still need to know that it will affect your team member. And knowing this will help you navigate managing the rest of the project.

When you think you understand how the change will affect your team, you need to understand how it will affect your clients or customers. This can be twofold, because your customer community is made up of leaders and individual contributors, each of which can feel very differently about the same change. The people you work with every day are usually the front-line employees who are most likely to be anxious about the change. They need to know that even amidst chaos, you have the project and the situation under some control.

Customer leaders are a different story. Executives may be betting their reputation or their career on this change of direction. Or this change may have been forced on them, and they expect to deliver it, even if they aren't totally sold on it. More important for you, if an executive was an important piece of your project, as a decision maker, an advisor, or a steering committee mem-

ber, then they will now be distracted. If they have time for the project, they will be much worse prepared and probably much less effective. You may not realize it now, but this is a bigger problem for you than the change itself.

■ **Note** One of the scarcest project resources is executive focus. Change can limit that in ways you don't expect.

Being aware of the impact of change on the people involved in a project is vital to your ability to run the project. These people include your team and the project's immediate customers. But the people most affected may be the executives or sponsors of the project who now have other priorities or other things on their minds. They may not be there for you and your project if you need them.

Be Aware of the Project

Finally, you need to understand whether and how this change impacts the project itself. You've already considered the people and the situation, but you haven't considered whether this change affects the way you need to run your project, or if you need to step back and redo some of your deliverables to adjust for the change that just came along.

Large changes come with a certain amount of risk by their very nature. But usually, you do risk analysis only for the direct change that is happening, not for its indirect changes to your project. This can bring many types of risks upon you. It can be schedule risk, where a change delays your ability to launch your project. Perhaps it's resource risk, in which the change consumes resources you need for your project to be successful. It could also be market risk; your project relied on certain market conditions that are no longer present when you're trying to deliver. Or maybe something else you need to add to your risk-mitigation strategy wouldn't exist if not for this change.

Even if the change doesn't carry a large amount of risk for you, it may require that you revisit virtually all of your plans and documents. Constraints or requirements may have changed. You may need to adjust your rollout plan or alter your testing strategies based on the new situation. If the change is big enough, you should consider reevaluating all the work you've done to date to make sure it's still valid and relevant. If it isn't, now is the time to find that out, not later in the schedule.

Keeping Aligned

Staying relevant isn't something you should take for granted. When you started out, you had goals and objectives, and you had a customer base with a problem that needed solving. You worked hard to come up with a set of principles that would help solve this need, and your team has been heading merrily toward this resolution, comfortable that it's doing the right thing.

It's important to determine whether all these things are still true. You need to work with your team and with the decision makers in your project community to make sure your project should continue down the path you're already on. Certain members of your project environment may feel a little deflated or discouraged due to the change, and in that case you need to remind them that the problem you're trying to solve still exists and your solution is the best way to address it.

You should be very visible and transparent in making the decision to continue on the same course. If your team or your clients think you're burying your head in the sand or otherwise ignoring what is going on around you, they will start to lose faith in your ability to lead. Do the work to make sure your project is still the right thing to do, and then confidently continue to lead it. And don't be afraid to get the opposite answer—maybe your project no longer makes sense. If so, go confidently in the other direction.

Summary

Change is a constant in life and ubiquitous in projects. You should expect it to happen: failing to plan for it will cause you to be a marginal or ineffective project manager.

The first key is to manage change intentionally, just as you do everything else. Don't try to pretend that it's not happening or that it isn't impacting you or your efforts. Before you can begin managing the change, however, you need to know what is driving it. The drivers of change are many and varied. The cause can be a combination of factors that necessitate a course correction. But knowing why things are happening will help you take the next few steps.

And those steps are to understand and become aware of how the situation, the people, and the project are affected. Note that each can be altered in different ways, and people on the same team may view this change differently—perhaps very differently. As much as you can, being aware of the new environment will help you and your team.

Finally, make sure your project makes sense in this new reality. Your project is probably still necessary, and you should continue to try to deliver on it. But it's also possible that the project you're trying to run is obsolete or no longer a priority. Either way, you should be doing this kind of analysis transparently and with clear direction. Whether you cancel your project or double-down and start working on it even harder, make the decision with a full understanding of the situation. You're probably doing the right thing.

Align with the Client

“Hmm ... I don’t remember talking about that.”

Let’s envision this fairly typical experience for a project manager. She launched her project a few weeks ago, finished the operations manual, trained the IT staff, and held a celebratory lunch. The functional manager has agreed that the project is closed and the hand-off deliverables are complete. In other words, the project manager is all done.

Catching her breath, she takes a two-week vacation someplace warm and sunny, and is now back at work, ready to pick up a new project. New projects are exciting and often come with that “new project smell.” However, even new projects are usually somewhat old and can use a bit of freshening up. Many project managers have been given a project that has a name with a year in the title—after that year has already past.

So, the project manager is having the kickoff meeting with her boss, who hands her a packet of information and starts talking about the project. Somewhere in the conversation is an offhand comment about “reconfirming what the business needs” and “seeing if anything has changed.” These kinds of statements are commonplace and often overlooked. But project managers who disregard such remarks do so at their own peril.

The project in this project manager’s hands may have been approved a month ago, a quarter ago, or a year ago. It may have been a really good idea at that time, based on what the reality was then and what capabilities existed

at the time. In today's fast-paced environment, virtually none of the assumptions may still be true.

■ **Note** Even decisions made yesterday can be out of date. All preexisting discussions should be considered suspect.

Before our hero can begin working on her project, she needs to make sure the goals of the project, the customers, and the project team are all very well understood and very much agreed upon. In fact, as discussed in other chapters, she should also reconfirm that the team, customers, and stakeholders are still the same as they were when the project was first approved. Failing to gain this alignment is not only a very obvious way to have a project fail, it's also a failure in project management. A project manager who attempts to deliver without full agreement—or at least full understanding—on what's being delivered isn't doing the job correctly.

Why Alignment Is Important

In just about every way, proper alignment provides a higher level of execution. If you consider an automobile with one tire slightly off—or even a grocery-store shopping cart that has one squeaky wheel—you can understand what misalignment feels like. No matter how good the other three wheels are, how new, how perfect, they're limited by the one that isn't. The person trying to push that shopping cart through the parking lot is in for a rude surprise.

When everyone on the team is *aligned*, which means literally pointed in the same direction, then the team can operate at a totally different level than they did in the past. The shopping cart now speeds downhill instead of careening off to the side. Complex issues become minor annoyances, and minor annoyances disappear.

Equally important is the impact of alignment on the team's morale and performance. When everyone understands the goals of the project and can draw a line connecting what they're working on to how it will help achieve one of the goals, something magical happens. Team members begin to work together better, they stay focused longer, and they think two or more steps ahead and consider how their component or whatever they're building will

be used. They start thinking as if they're going to be a permanent resident of the new world you're building, and they act accordingly.

Alignment can even make contentious discussions much more calm and focused. Your team anticipates each other's needs, simply because they're thinking about them earlier. Even when they don't, they work on shared problems together. There are fewer statements such as, "I built my component to spec; your requirement wasn't documented anywhere." When everyone on the team has the same goal, they feel like a team and stop acting as individuals.

■ **Note** A bunch of people heading in the same direction together is a team. A bunch of people heading in the same direction individually is a race.

Alignment with the customer¹ is even more important. A customer who feels aligned with the team will also be able and willing to help when problems and differences arise. If everyone wants the same thing, and everyone understands each other's needs, then figuring out and negotiating details becomes much simpler, and large amounts of drama are removed from the process.

Aligning with the Client

Before you can gain alignment with the project team, you first need to understand the goals and objectives of the project. And before you can start that, you need to understand where and how the project fits into the organizational strategy.

Consider Figure 6-1, which illustrates the strategy pyramid. At the very top is the long-term strategic vision of the company. This is the kind of thing that rarely changes and can take several years or decades to achieve. Below this level is what is called *strategic direction*. This is a list of near-term strategies that may take five or more years to reach fruition.

Below this are the projects that help support the strategic direction. As a project manager, this is where you come in. You're given a project that

¹ Throughout this book, I use *customer*, *client*, *business owner*, *product owner*, and similar words interchangeably. They all have the same meaning: the person, persons, or representative who is telling you what they want done.

maps to one or more of the company's strategic directions, and it's up to you to break that down into discrete tasks that must be achieved to be successful. These tasks are what make up the bottom layer.

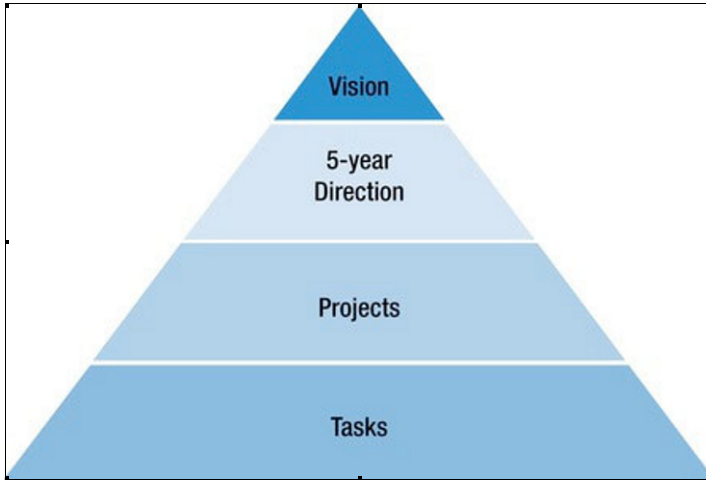


Figure 6-1. The strategy pyramid

So, how do you get started? The best way is simply to ask. Chapter 8 talks about stakeholder identification. Now is the time to interview the group of people you uncovered. Not only does doing so get you the information you need to gain clarity, it also helps show the community that you're listening and that you care about its needs and objectives. This is true even if you decide to ignore their thoughts because they're different than those of other stakeholders. At the very least, you can find out that some people disagree, and you can inform them that they won't be getting what they want out of the project. Better to say it now than to have them find out when it's just about time to deliver. Of course, the importance of each stakeholder is different, as you see later.

The goal is to get people talking but not lead them too far. You should act as if you know nothing about the project and are learning the possibilities via this conversation. And you should act that way because for the most part, it's the truth. Here are some interview questions you can ask:

- *"How does this project assist in furthering the company's strategic direction?"* Virtually everyone you ask will have a different opinion, and the responses will range broadly. This is not only normal, it's desirable. It helps give you the full spectrum of choices for what the project could

be. Ask for reasons and details about the answers people give. You may find that some of your stakeholders are unaware of the vision and goals of the project—or even that they disagree with them! The more people you ask, the more you’ll find commonalities and themes. You use these a little later.

- *“What do you hope this project delivers?”* Achieving the strategic direction will most likely require several projects over several years. Your project is usually just a single part of it. With that in mind, what do the stakeholders hope you can achieve? If the responses seem very aspirational or even impossible, now would be a good time to try to reset their expectations. You can do this with a question like, *“Do you think we can accomplish all that within the scope of this project?”* It’s difficult to ask this without it sounding like a leading question, but usually that reminds them that projects are temporary, and strategies aren’t. Your goal is to take several steps in the right direction, but rarely does a project get you all the way to the end.
- *“What do you think other people hope this project will deliver?”* Coming at the question from a different angle helps uncover conflicts, grudges, or past negotiations that had nothing to do with you, the project, or the goal. Your operations manager will use their answer as a way to warn you about demands of marketing, and marketing will tell you that the IT group doesn’t understand the business. Each group reveals a little about themselves in the process and gives you a hint about where conflict points exist so you can avoid them.
- *“If this project does only one thing, what do you hope it is?”* Asking each stakeholder this question helps you get alignment and focus on what’s really important. It’s very important to focus on the “one thing” and not accept a list of several. This kind of question is very powerful and should cause your interviewee to think. Along the way, you’ve been asking similar questions and getting a verbal bullet-point list of what the stakeholders hope to get accomplished. These lists are unlikely to be identical, but they should overlap at least a little. When you ask for one-item answer, some groups have the same top priority, whereas others have very different priorities. This is exactly what you want to discover.

■ **Note** Differences in alignment before the project begins are okay and do not necessarily mean conflict is coming.

The following sections show what you do with all this information; for now, it's enough just to have it. Without this information, you can't proceed. But you need to do more with it before you can truly claim to be aligned on the project aims.

Aligning with the Team

By now, you should have a sense for what the business, client, or customer hopes to achieve from the project. Now it's time to ask your team members what they want. This list is very different than the list you already have in hand, and that is okay and perfectly normal. Keep in mind that your client community is interested in the outcomes—they want you to build or do something that they can then use to achieve some other goal. In other words, to your client, the project isn't a goal in itself.

Your project team, however, probably won't use the project's final deliverable. Team members are much more interested in the process, what new skills they may acquire as part of this project, and how this will impact their career and what they do next. These desires may seem a little selfish, but this is the point in the project when it's okay to be selfish. You need to let your team members express their desires now, before there is a deadline, and before making decisions for the good of the team or the good of the project becomes necessary.

■ **Note** Remember that your project team is made up of people who have needs that go well beyond the scope of your project.

Your project has two separate sets of goals: those of the project and customer, and those of the team members. Attempting to manage while only knowing one or the other will leave you with a disappointing result. Worse than that, without knowing what your project team hopes to accomplish, both separately and individually, you can't understand people's motivations and how to keep morale high throughout the lifetime of the project.

How do you, as project manager, get this information? The same way you got information from the client, although you need to handle the process a little differently. Depending on the size of the project, you should meet with the key people on the team before the work begins. If the project is too big, or parts of it are outsourced, or doing so simply isn't feasible, then ask yourself, "Which of these people do I want to work with again?" This list of key people may be long, short, or equal to the number of people on the project. However, under no circumstances should it be zero. If you don't think there are any key people on your project with whom you hope to work again, then you're missing an opportunity to create a long-term relationship. Even if your project seems to be full of team members you'd rather forget, there's a good chance you'll see them again elsewhere. Before the work begins, take time to interview your team, just as you did your customers. The questions are similar to the ones earlier, but slanted differently:

- *"What are your hopes for this project?"* This is the easiest question and a very interesting way to get someone talking about their goals, without having to give them a direction. Some team members will talk about the project itself and say they want to make an impact or reach certain financial targets or Key Performance Indicators (KPIs). Others will say that they would love to work on a project that launches on time, or on that even launches at all. Other responses will range from a desire to travel to the client location (or a desire not to), to hoping to be finished before they go on vacation, to any number of random desires. What's important here is that they're telling you what's important to them. So listen and pay attention. You may not act on all the responses (in fact, you probably won't), but you do need to know what they are.
- *"What do you hope to learn from this project?"* Everyone has something they can learn, and most of your team members have certain skills, technologies, or arts they'd like to either gain or improve. This can range from presentation skills through the newest coding language. You need to know these desires for two reasons. First, if people are hoping to learn something that your project won't provide, it's best they learn that now, not half-way into the build. Second, you, as project manager, need to be looking for opportunities to give people what they're looking to do. If you don't know which team members want to learn more user interface programming, and you have UI tasks to dole out, you miss a great way to motivate and reward your team.

- “What do you think you’re best at?” This is a trick question, if done subtly. You want to let your team members tell you how great they are, and you should make sure you’re listening and reacting. People answer with resume bullets, or a description of their passions, or both. This is useful information and exactly what you want. Just as when you asked people what they hoped to learn, you can use this feedback to give work assignments, to figure out whom to turn to when there is a problem or a schedule delay, and whenever you need an expert.

Projects are often the victim of turnover; the project and project manager remain, but the team is constantly changing. This is increasingly true the longer your project runs. When a new person joins, you should ask them these questions right away. Turnover may happen at a bad time: a new member may come on board during a crunched release or right before an executive update. But this meeting has a very high return on investment. Skip it only in the most extreme circumstances.

Just as you need to get to know who your clients are and what they want, you need to get to know your team members and what they want. You also need to recognize that these lists at best overlap, but more likely contain two entirely different sets of goals. It’s your job as project manager to figure out how to reach the project goals while maximizing the number of team goals you’re able to deliver. Your team will understand if you have to sacrifice a team goal for the good of the project, but they won’t forgive you if you miss a chance to achieve one for no gain—and they will probably feel even worse if you never found out their personal goals in the first place.

Communicating Alignment

There comes a point in the project when you’ve talked with everyone about what you’re going to be building and, more important, what you aren’t going to be building. You also have a handful of technical requirements, restrictions, and constraints that you must navigate in the course of delivering the project. You probably have more of each than you can possibly convey in a simple manner. But you must communicate this information to your team, stakeholders, and management at minimum. Depending on the project, this list may also include vendors, third parties, regulatory bodies, legal, public relations, compliance, and a list of groups too lengthy to fathom.

You must get the information out to all these groups in a way that doesn’t require dozens of hours of your time, that isn’t ambiguous, and that doesn’t

require constant changing and altering. These statements of alignment are called *project principles*. They're statements about the project that clearly define the direction the project is taking and highlight any key constraints or obstacles. These aren't meant to be requirements statements or wish lists.

A good principle should be clear: there aren't multiple ways to interpret what it means. It shouldn't use *code words*—words that have a different meaning to the project team than to someone who hasn't been closely following the project. It should seek to address everyone's top needs, even at the expense of some other needs. A healthy balance between goals and constraints is a good thing, and you need the customers to see that their needs are being addressed. At the same time, the team and other project stakeholders need to be certain that you heard them.

■ **Note** Project principles should cover the entire project, and not just one stakeholder group.

More important, these project principles should rarely, if ever, change. They should endure for the length of the project, even if requirements, team composition, or stakeholders change. If something happens that causes you to violate, change, or otherwise ignore these principles, then that is a sign that you have a material change on your hands, something that may change the project's direction completely.

Once you have this list of principles, you need to distill them down into the following parts:

- The principle itself: a single sentence that asserts the statement of principle
- Why this principle is important: one or two sentences about why this principle is on the list, and why it matters
- Risks that this principle introduces to the project

Let's consider an example. Imagine you're a bearing manufacturer, and you currently make 3"-diameter ball bearings. Your project is to make 3" cylindrical bearings to sell into the same market. You could have a principle that looks like this:

- *Principle:* We will use existing machines, lathes, and equipment to manufacture the cylindrical bearings.
- *Why this is important:* The business case is predicated on not having to buy new equipment to manufacture this product. Our staff is

trained on the machines we have, and floor space in our manufacturing facility is expensive.

- **Risks:** Our existing equipment might not be appropriate for the manufacture of this item. Alternately, our existing equipment may create a substandard product.

These five sentences are all you need to express your statements of principle. Anyone, no matter what function they represent, can clearly understand the decision that was made, why it was made, and the implications of the decision. Even if you disagree with the direction, it's difficult to claim to not understand it and base all future decisions on it. Remember, alignment doesn't necessarily mean everyone is happy with the direction; it means everyone understands and accepts the decisions that have been made.

Understanding Differences

As you go through this process, you'll find differences of opinion. This isn't unusual, but there are scenarios in which you, as project manager, need to address them. The longer you let the project run in a misaligned state, the harder it will be to bring it back on course. You'll have the urge to figure things out as you go or worry about issues as they come up. Try to resist this whenever possible, especially with foundational, goal-affecting issues. You'll have differences along the way, such as, "Should the new bearings be silver or black?" This kind of difference can be postponed. However, if you run into someone who doesn't think the goal is to sell as many bearings as you can, then you probably have an issue.

There are two main flavors of differences: *interdepartmental* and *intradepartmental*. *Department* in this case can mean division, function, geography, or even company if you're managing a multicompany project. The main idea is that sometimes groups don't agree with each other, and other times a group can't seem to agree with itself. Both situations present very different challenges and can cause different kinds of disruption to your project along the way.

Interdepartmental differences tend to be caused by groups that have very different goals. A classic example is the operations department, which wants to have a stable, supportable capability, and the business, which favors speed to market over everything else. Both groups are acting correctly based on their own needs and goals. Solving these differences usually involves straightforward negotiations about project scope, project timing, or something else. Assuming that in the end all the key decision-makers do want to sell

more bearings, you'll eventually come up with an answer that everyone can agree with.

■ **Note** Although you should try to meet the needs of as many stakeholders as possible, you needn't meet them all at the same point in the project.

Intradepartmental squabbles are a much different problem. A lack of internal alignment is often enough to pause, delay, or even cancel a project. This kind of disagreement comes when you translate the goals of the project with the strategy. Some stakeholders agree with the strategy but disagree that this project is the right way to make progress toward it. Or they disagree with the validity of the constraints, or they think the risks going into the project are too great. Sometimes they disagree just to be disagreeable. The root cause isn't as important as recognizing when you have a lack of internal alignment.

One thing to do when you find yourself in this situation is to go back to your executive and stakeholder interview notes. Perhaps you and the project team mistranslated the original goals into the list of tasks and principles you're taking on. Conversely, perhaps the stakeholder or stakeholder group who doesn't agree doesn't have the same context about the project you have. Either way, this is why having this information is so important. To be able to draw a top-down line from strategy, to interview, to project, to project task creates a very clear picture.

Unfortunately, this often isn't enough to get people back on the same page. So, you need to gather even more information. One way is to interview the person or group that is in disagreement. You can start with a bold question such as, "Do you think we should halt this project?" Note that you need to resist the urge to move ahead with this disagreement looming; often, forcing this kind of yes-or-no question can help frame the next discussion. If the answer is that yes, you shouldn't begin, then you have a clear task. You need to find the person or group who approved the project, and a few others, and see if the objection is real and worth changing the course of the project and portfolio. More likely, the answer will be, "No, you can get started, but ..." Here's where you need to be very careful and are likely to get yourself into trouble if you do move forward.

■ **Note** It can be tempting to move ahead without alignment. Don't.

At this point, you can try to get confirmation about the success criteria you've collected, look at the "one things" your interviews indicated people want to achieve, and see if you can bring the group back into alignment. If the objection isn't strong enough to halt the project, then there probably is room to work things out. There may be success criteria that some people didn't know about, or they may disagree with the criteria you've collected.

This situation isn't uncommon, and a good project manager should be ready to deal with it. Continue to communicate your top-down distillation of requirements and goals, and resist allowing new goals to come in from the sides or, worse, bottom-up. Although doing this to the best of your ability may not be enough, not being prepared and not being ready to have this discussion will certainly cause you not to succeed.

Amount of Aggressiveness

An old chestnut talks about under-promising and over-delivering. This is the ideal, but it's particularly difficult to do. Your client, and probably your Project Management Office (PMO) or program manager, knows that if you deliver everything you promised and nothing more, you probably weren't being aggressive enough. Because you know that work expands to fill all available space, if you complete all your tasks, there is nearly always room for one more. But if you promise more than you eventually deliver, you look like a poor project manager, and clients are disappointed at what they get. What's a project manager to do?

Earlier, the book talked about managing scope and how important it is to have more than one in/out line. Items that are below the out line should stay out. They may be removed for lots of reasons, only one of which is availability of resources. However, you still have varying degrees of in items, and this is where project managers can reach for a little more while still setting the proper expectations.

The discussion of project principles left out *project-management principles*. These are the principles that exist from project to project and are more applicable to you—the project manager and your portfolio team—than they are to the task at hand. One of these project-management principles should always be, "Everything not in the A list is at risk of not happening." This is a

statement you should repeat at every meeting, after which you should wait for a sign of acknowledgement.

However, you should also plan to deliver as far down the list of requirements as you possibly can while still maintaining the level of stability you need for your final delivery. Although you're saying, "We're only doing the As," you need to create credibility with the business or client for the next project that some of the Bs and Cs will get done. This is especially helpful if you run into issues and need to pull all the Bs and Cs off the table for a certain project. You at least give the appearance of trying, which is just as important as actually delivering for gaining and maintaining alignment.

Remaining Aligned

Finally, you must have a plan to keep your team and your stakeholder community in alignment as the project goes forward. This activity is separate from tracking progress, status updates, issue logs, and so on. It's a constant reminder of what the project is doing, what its goals are, and what everyone will see at the end. Your project principles should help with this.

As you move forward in the project, be certain that your principles remain both true and active. If you violate the project principles, then you generate confusion among the people who agreed to them in the first place, and you may start to get requests to violate more of them for random reasons.

Remaining aligned through the life of the project is difficult, especially as conditions change, team members change, and the market changes. But you have a simple mechanism to keep up: use your principles, keep them alive, and keep communicating them. You know you've done it enough when everyone is tired of seeing them and knows them by heart. That's the sign you're looking for.

Summary

Gaining proper alignment is vital to the project. You need everyone to be heading in the same direction, even though the members of the project team may all have different goals. The easiest way to do this is to map your tasks to your project, and your project to corporate direction, and the direction to the overall vision of the organization. You can distill all this into a handful of principles, which you can then use to communicate clearly and easily with all groups. Once you have agreement on the project, you must

keep it that way with consistent, accurate reminders of the project principles. And, as project manager, you must stick to your principles.

If, midway through a project, you tell your portfolio manager or PMO, “My stakeholders don’t know what their goals are,” you aren’t likely to get much sympathy. Disputes over requirements, ideas, or individual tasks are fine, but disputes over goals never are.

Testing Assumptions

“Well, when we started this project, we were assuming...”

Life runs on assumptions. Every day, you assume that your alarm clock will wake you up, that your shower will have hot water, and that solar flares won't fry the electrical grid and broadband network that power your life. In fact, you couldn't live without assumptions; if you needed to verify each little thing every single day, it would be bedtime again before you were certain that everything you were assuming was true could be proven in the affirmative. And as luck would have it, a huge percentage of these assumptions do turn out to be true, allowing you to live in relative constancy.

Projects are very similar. In order to get started on a project, you need to begin with a certain number of things decided (or at least pretended to be decided). If you go ahead with nothing settled, then you're inviting disaster along the way. However, if you wait until you have enough information and details to make all these decisions, then the market and the project will pass you by. You have to make intelligent guesses about facts not yet in evidence. These are facts that either you have a reasonable expectation are true or can be made true through the course of the project.

The Project Management Body of Knowledge (PMBOK) tells you to document all your assumptions up front and get them approved and signed off

on. This is all very well, but although it somewhat covers you, the project manager, in the event that one or more assumptions turn out not to be true, it does nothing to help the project succeed. What you really need to do is make your assumptions up front, figure out how to test them, and then either retire them from the list of assumptions or turn them into full-fledged risks. The more time passes, the less acceptable it is to have and make assumptions. Part of the job is to learn along the way, determine whether assumptions are real, and decide what to do about them. No project manager gets much credit for identifying assumptions, because that's less than half of the job.

■ **Note** Merely identifying assumptions is almost never enough. You must do something about them, too.

If you work in an environment where failure is fine as long as it's fully documented ahead of time, then the PMBOK advice is probably your best bet. Even if you don't work in such an environment, it's always good to document and create visibility and plans around the assumptions you're making. However, if that is all you do, then not only are you asking for trouble, but you're asking for either your client or your program manager to come knocking on your door with a well-justified feeling of disappointment in you.

This puts a project manager in bind. You must have assumptions to kick off your project, but you can't rely on them to save you if they turn out to be faulty. If you don't constantly manage them, they can blow up on you and sink your project. And although you may not suffer for them personally, your project will, and no project manager wants that to happen on their watch. With that in mind, let's look at assumptions and how you can best use them. You can't live without them, so you may as well learn a little about how to turn them into a positive for the project.

What Is an Assumption?

Let's start with the easiest question. An *assumption* is a statement of fact without enough supporting evidence to prove its veracity. Assumptions mostly come into play during the initial planning stages of a project, but they can arise all the way up to the launch date as new information and evidence are uncovered. For the purposes of the project, you decide that a certain

statement is true, and you proceed with planning the project as if it were proven to be true.

Let's go back to the bearing example from Chapter 6. You have a project principle of "Must use existing equipment." This principle includes some descriptions and a risk analysis. Baked into that principle, but left somewhat unstated, are two assumptions:

- The existing equipment can create the bearings you need at the level of quality the market requires.
- Your current staff's training on the equipment is applicable to creating this new product with the existing methods.

Someone reading that principle can probably pick up on these assumptions, because they're either explicitly called out as risks or they're stated in other words. Because you've decided that this constraint is real, you need to make some decisions about the assumptions. You can't state that you'll get the best equipment for the job, nor does it seem you have the ability to re-train your manufacturing team.

Additionally, there is no way for you to test that both assumptions are true until you get far into the project. You probably won't know the quality of the product that comes off the line until it actually starts coming off the line. And although you may pilot production with a few well-skilled team members, it's unlikely that you can give everyone a test run in order to make sure they're properly trained. Nor is it necessarily true that the skill of your pilot members is representative of the staff at large. This single principle causes you to make an awful lot of decisions without the information required to know if you're correct. In other words, you have to make some assumptions.

Before you can make any assumptions, however, you need to do at least a little due diligence on the topic. This includes doing market research, or learning or understanding industry best practices. It can mean adding a subject-matter expert to the team or calling a consultant who does this kind of thing for a living. If none of those options are available, you may need to hire someone who knows the space to be part of the project. What it doesn't mean is guessing or making ill-informed statements. Having someone from accounting make assumptions for manufacturing, or having someone from manufacturing make a decision about sales forecasts, is likely to lead you down the wrong path and eventually cause someone to object.¹ Even if it does turn out to be correct, an assumption made without any diligence is

¹ They may even object if the decision turns out to be true, just on principle.

better termed a guess. And guesswork rarely factors into a well-formulated project plan.

■ **Note** The level of due diligence you perform on an assumption keeps it from being nothing more than a guess, and perhaps an uneducated one at that.

This brings you to the final important point about assumptions: they must be highly likely to be true—and 51/49 isn't nearly good enough. The bar you're trying to reach is that when an assumption is proven false, many people, including subject-matter experts, are surprised. An assumption must be something that most people, on first read, assume is true, and this should include people who truly know what they're talking about. Failure to perform this level of investigation before listing an assumption indicates to your project team, your client, or your managers that you took a guess and had no confidence in going forward with what you had.

The Importance of Assumptions

Even with these caveats, it's difficult to overstate the importance of assumptions in life. Some of the assumptions you make in your daily living are truly life or death, such as assuming that people will stop at red lights and won't push you into the path of a speeding train. Although your project assumptions may not get anyone killed, they may mean life or death to the project itself. Many assumptions can cause projects to be halted if they turn out to be false; or, looking at it another way, if you knew some of your assumptions were false when you started, you might never have approved the project to begin with. As a project manager, you never want to be working on a project that was approved on false or faulty premises: for a while—perhaps even for a long while—everything appears to be fine, but there is such a fundamental flaw that no amount of hard work or cleverness can overcome it. Don't put yourself in this situation.

This is why it's so important at the beginning of a project to have a list of critical assumptions that are highly likely to be true. Critical assumptions may include such things as, "A vendor can be located that provides the required materials we need at an acceptable price." Or, "Customs regulations will allow this kind of cross-border transaction." These are added to the list of assertions on which the project is founded. (Assertions include the key

assumptions, constraints, and value positioning of the project.) From these assertions, project principles, and requirements, forecasts are built. Without them, you can't have a project.

You need a problem statement, as well as an idea about how you're going to address the problem, to have a project. Assumptions help with both. They help define the problem, and they help define the intended solution. Without them, you don't have enough to know where to begin. And a project manager needs to have a high degree of confidence that the decisions being made are based on things that are either certainly true or at least probably true. For instance, before you can begin a project that has an assumption about the availability of a vendor, you should do enough research to know that such vendors exist. When you start making plans based on things that *might* be true, you wind up swapping your role for that of a fortune teller.

■ **Note** Listing assumptions is no replacement for getting a good answer from the subject-matter experts or impacted function.

It's important not to let the use of assumptions act as a shortcut for communication. It can be tempting to list certain things as assumptions without checking with the person, group, or function that could determine the proper course of action. When you as project manager begin passing around a list of assumptions that include things from other groups, your boss will ask, "Did you ask [person x, y, or z] about this?" Woe be unto the project manager who begins listing assumptions without at least asking a few simple questions. Your guesses are mere wishes. A wish implies that you were afraid to ask, because you knew what the answer might be. That is no way to begin and no way to instill confidence in your project team.

Assumptions are important to the project. Indeed, the project almost certainly won't be approved without them. It's the project manager's job to understand what assumptions are made when the project is approved and to verify them. There are things on the list that can't be known until well into the project, but it's the project manager's role to move each one up the chain from wish to guess to well-reasoned assumption, and finally to either something that's proven true or a risk that must be managed.

As you can see in Figure 7-1, the more due diligence you do, the more you're able to move an assumption higher up the value chain. This doesn't necessarily mean that investigation will prove an assumption to be true.

Turning an assumption into a risk should be seen as good project management, not the reverse. However, even assumptions that turn out to be false need to be managed and may not end a project. Managing risk is something every project manager should be accustomed to and should have well-documented plans to address. Managing the unknown is a much more difficult task; managing things that you're intentionally leaving unknown is simply inviting trouble when there is no need. From this perspective, "known unknowns" are little better than "unknown unknowns," and they may be a sign of something else.

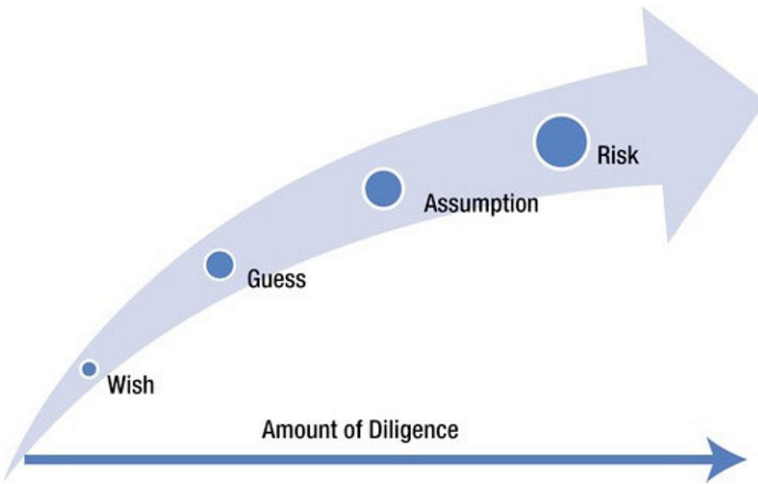


Figure 7-1. The lifecycle of an assumption

How to Identify Assumptions

You have to worry about two different classes of assumption: pre-project assumptions and assumptions made by the project team. They each come from different sources, and represent different kinds of risks and challenges.

Pre-Project Assumptions

The first are the ones put together before the project was approved. They're usually pretty up front in the project approval and are easy to find. They appear as constraints, principles, or some other correct-sounding assertion, but they rely on unproven information. These assumptions are usually formulated by people who are well-intentioned but very much out of their depth. These

kinds of assumptions are usually the easiest to manage, because whoever put them in usually did very little investigation about them ahead of time. Some of them will seem silly, much like, “Assuming I can still eat chocolate cake, I will go on that diet.” These types of assumptions really aren’t assumptions at all, or at least not in the traditional project sense. They can often be managed through relationships, collaboration, and other means of influence. “C’mon, you knew you weren’t going to get away with eating all that cake. What did you really mean by that?”

Examples of these kinds of assumptions are, “We can open an office in New York City without being subject to New York state sales tax.” Or, “The rockets we need to build are narrow enough to fit through railroad tunnels.”² Probably no one checked these things, but common sense tells you they’re probably true.

Occasionally, you run into a pre-project assumption that someone either believes strongly about or holds dear for some reason or other. You mostly find this when you’re dealing with executives who have been removed from a function or information for a substantial period of time, but still feel they know the way things are run and are current on the best practices and limitations of the function. This is trickier to handle than the cake example, but the solution exercises the same muscles. I once worked with an executive who had a deep background in call centers. One of the key constraints from back in his time was that the world was running out of 1-800 numbers, and that those numbers were precious commodities. He was adamant that our solution needed to conserve the use of these numbers, because they were expensive and hard to obtain. Of course, since he left that function, 888, 877, and 866 numbers have been added; and although they aren’t infinite, they’re much less limited. The executive nearly placed a constraint on the project that wasn’t real at all.

Unless you happen to be an expert in the same area, you can easily position yourself as a neutral party. In fact, this is preferable—you aren’t on anyone’s side; you’re on the side of the project. If you’re able to put yourself in this situation, then you probably find yourself between an executive on one hand (the one who knows how things used to be) and the subject-matter expert or experts on the other (the ones who know how things are). From here, it’s a fairly simple conflict-resolution exercise; and more often than not, the issue is resolved without much of your involvement. There is either an information gap or a leadership gap that needs to be repaired, but both of them should exist outside the charter of your project. You have two key

² www.snopes.com/history/american/gauge.asp.

tasks: making sure the conflict is resolved somehow, and making sure you're aware of the final outcome.

In the 1-800 example, once we understood that the executive was working from a belief that was true years ago but no longer true, we were able to walk him through why we didn't need to worry about his constraint. It was a very easy conversation to have. It wasn't that he was wrong, it was that things had changed around him.

The pre-project assumption is the easy type. It's usually well-intentioned and made by someone who either wants the project to succeed or needs a little more education on new realities. Both of those types of people can easily be a friend to the project and, by extension, to the project manager.

Assumptions Made by the Project Team

The second type of project assumption is much more difficult to spot. These are assumptions made by the project team itself after the project is under way. They aren't made by optimistic executives but by normally cautious subject-matter experts. Often, these folks offer statements as if they're verified, even when they aren't. Because you, the project manager, don't have the level of expertise they do, it can be very easy to accept such statements at face value and move on. In so doing, you wind up building your project on unverified statements that you haven't captured anywhere.

I once witnessed a project that was essentially a complete rewrite of the company's order process. The team made the assumption that the only group that needed to be involved was the ordering team, and they convinced the project manager they were right. The project manager, being thorough, had the names of the other groups on the list of stakeholders, but due to the information gained from his team, never contacted them. When the new process went live, the order-processing team was up and running, but everyone else was down. The chief designer simply shrugged and said, "Huh. We never knew."

This kind of problem usually comes up months later, when you're facing a problem and ask the expert who made the initial statement something like, "But I thought you said ...". This is usually followed by the person shrugging sheepishly and admitting they based their statement on unverified information at the time—information that turned out to be wrong. The role of the project manager is not to be proficient in all areas, but rather, to be proficient in understanding how to manage these kinds of people and how to determine their level of confidence in their statements.

■ **Note** Project managers don't need to know everything, but they do need to know how to communicate with experts.

Level of Confidence

The process of determining level of confidence isn't all that difficult; the problem becomes knowing when to do the questioning. There are obvious signs that someone isn't totally sure of themselves. They can include the phrases "I'm pretty sure that ..." and "I think that ...," or the answer may come haltingly or after a pause. If you've been managing people and projects for a while, you can recognize the signs of someone saying something they aren't completely sure about. But what happens when someone states something confidently and clearly, but in their heads, they realize they need to check something before they can be sure of the answer?

You need to ask three different types of questions at this point:

1. *"How sure are you?"* Ask this when you sense someone is unsure. Often you get a fairly weak answer like, "Pretty sure." Keep probing, asking what they aren't sure about, what information they need, and who they need to talk with. Sometimes that person is you or someone to whom you have ready access. Other times it's that person's boss, or a co-worker, or someone more senior in the organization. It could also be that the person is certain, but they need to look up one fact or single piece of information. However, they may be taking a wild or uneducated guess. This is okay as long as the person knows they're doing it and that they're on the hook for closing the certainty gap.
2. *"You sound very sure of that. What makes you so confident?"* Ask this kind of question when a person states something with no sense of uncertainty in their voice. A person who really knows has a very good, very quick answer. It often takes the form of, "We did this before," or, "I just looked this up for another project," or something that sounds like they did their diligence before talking with you, even if accidentally. But if a person is using confidence to mask uncertainty, this second question (or sometimes, a follow-up) leads them back into the "pretty sure" type of vocabulary. If that happens, you know exactly how to handle it.

3. *“Of all the assumptions or decisions we made today, which one do you feel least confident about?”* The two previous types of question are in-the-moment exercises. But you need to do one more thing at the end of meetings and at regular intervals along the way: ask what people are uncertain about. This can make people question their own confidence, or it can cause someone else on the team to ask for further verification, either of which can cause one of the two previous types of discussions to take place. During regular project meetings or status updates, you can ask, “Which of the tasks ahead of you are you least certain about?” Doing so can uncover decisions that are vague or simply wrong. People are generally unwilling to say that someone else is wrong; they’re more willing to say that they’re unsure about something. This can cause assumptions to surface that are trending closer to false than true, allowing you to manage them.

Whenever you’re coordinating a meeting in which decisions are being made, you should be on the lookout for the confidence level people show in their decisions. This is particularly true when the impacted function isn’t represented in the meeting; but representation doesn’t always mean you have the right representative, or that they will give you the right information. Use your own abilities to figure out how confident people are in their statements, figure out why they feel confident, and double check as often as you can.

Avoiding Standard or Implicit Assumptions

Listing and tracking your assumptions is important. However, there is an entire class of assumption that shouldn’t appear on your list: the assumptions that are always valid for any project. These are mostly assumptions that can be cut and pasted from a past project with only the name of the project changed. Examples of this kind of assumption include the following:

- Developers will be assigned and staffed when needed.
- Budget for project will remain as stated in project charter.
- Subject-matter experts will be available for consultation on an as-needed basis.

These assumptions aren’t very useful, and listing them in your project will never help you. If your development team gets pulled to do something else, having these assumptions won’t change anything. Telling your client or your program manager that in order to do 2,000 hours worth of development, you need 2,000 hours of developer time only shows that you know how to add. It doesn’t show you managing anything; in fact, it may show you abdicating the management of the project.

If you start micromanaging these kinds of assumptions, you stop managing the project. These assumptions aren't deliverables, and they shouldn't be treated as such. Sure, you need 2,000 hours of developer time in order to deliver. But what if you only get 1,990? Does that mean your project will fail? What happens at 1,600 or 1,200 hours? Can you deliver on a portion—or maybe just the highest-value requirements? What happens if your budget gets cut or a key expert is assigned elsewhere? Are you willing to state ahead of time that this will cause your project irreparable harm? In so doing, you may limit your own management ability before the problem even occurs.

■ **Note** Some things that look like assumptions are actually risks that you need to actively manage.

The fact is, these aren't assumptions: they're risks. And many of them are the project manager's responsibility to solve. A project manager who says that a project failed due to the lack of availability of an international tax expert, even though the expert's availability was clearly called out as an assumption in the project documents, won't be greeted with kindness by the program manager. You don't get to reduce your responsibility to manage project risks by assuming them away. To do otherwise is to stick your head in the sand and hope for the best. As pointed out elsewhere in this book, hope has no place in a project plan.

Assumptions vs. Risk

How do you know if something is a risk instead of an assumption? When you list an assumption, see where it falls in Figure 7-2. On one axis, plot your ability to influence the problem. On the other, plot resources under your control to help overcome or work around the problem. There are things you have no ability to influence. For instance, you may make an assumption that no new legislation will be passed in your subject area for the life of the project. Impacting legislation is usually outside of the scope of project managers. And you may have no flexibility regarding some aspects of the solution; sometimes either you are compliant or you aren't. But these kinds of problems aren't the norm. More frequently, either what you're afraid of is within your ability to change, or reasonable alternatives exist if you need to alter course. For instance, your new product may not be a success if it's only marketed to people without cell phones. Ideally, you have the ability to influence the marketing campaign even if you aren't in control

of it. In many ways, the ability to influence the course of events is at least as important as being in control of them completely.

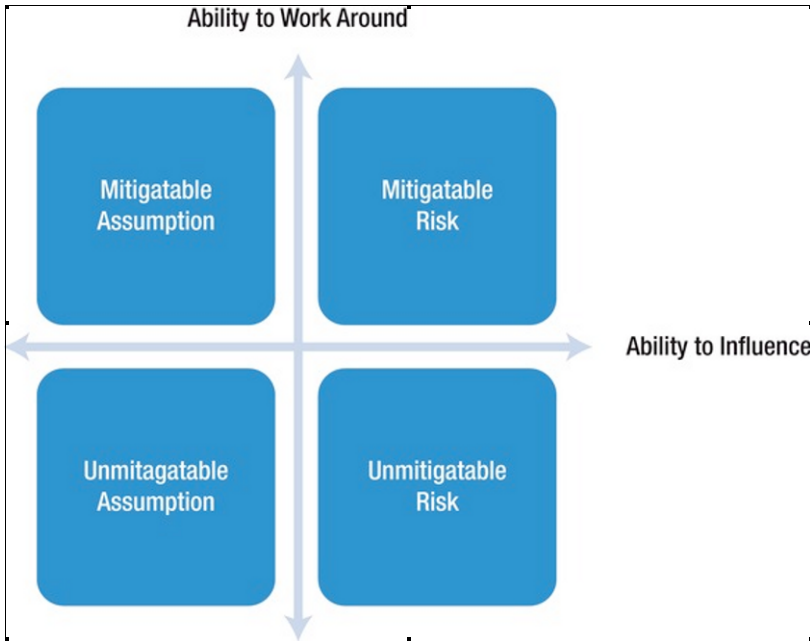


Figure 7-2. Types of assumptions and risks

In your four-quadrant diagram, only the items that fall at lower left truly are assumptions. These are things that you, as a project manager, can't hope to change, and if they don't go your way, your project is probably materially harmed. In the upper-right quadrant are things you can impact before they happen, and if they do happen, you have options for how to recover. They are mitigatable risks. In the lower-right quadrant are unmitigatable risks. You may be able to prevent them, but if they happen, there's little you can do about them. Finally, in the upper-left quadrant are assumptions that are mitigatable. Something may happen beyond your control, but if it does, you have ideas for how to work around it. This quadrant may involve more time, more money, decreased scope, or something else, but you can at least move forward with the project.

■ **Note** Determine whether the issue you're facing is preventable by you, and if not, what level of flexibility you have in overcoming it. This helps you understand if you're facing an assumption or a risk.

Knowing where your issues fall is very important when you need to explain them to your client or program manager. Although there are problems that you can't do anything about and would be fatal to the project, many issues are manageable. It's important that you, as a project manager, know where each of the problems falls on the spectrum so you can talk confidently about how, and whether, you should proceed. If you don't make this determination ahead of time, your program manager will attempt to move everything into the upper-right quadrant and tell you to figure out a solution. You need to have a firm understanding of the issues before they occur. Failure to do this is a failure in project management, not simply an unlucky break.

I once had a project in my portfolio that was completely at the mercy of the New York State legislature. There was almost nothing we could do to change the trajectory of the legislation they were about to pass, and if it went against us, the project would have to materially change. In terms of Figure 7-2, this was a lower-right risk. I asked the project manager what we would do if it didn't go our way, and he said he was considering some options. This was great news, because it moved the project from lower-right to upper-right and made it something we could work with. The problem was, the project manager was wrong. The team had no options, and there was nothing we could do—the new law sunk the project. By not considering this ahead of time, the project manager gave false hope, and we wound up with a bigger failure than necessary.

Continually Testing and Tracking Assumptions and Risks

You have four classes of assumptions and risks:

- The kind you can do nothing to prevent: if it happens, you need to rethink your project
- The kind you may be able to stop, but if you aren't successful, you're once again stuck
- The kind you can't prevent but can do something about
- A project risk that should be managed as such

Along the way in a project, you make lots of assumptions. Usually, this is due to imperfect information, lack of representation in decision making, or a lack of understanding of the problem. Whatever the reason, you have a list of things that fall somewhere along the spectrum of Figure 7-2. As project manager, you need to continue to move your assumptions up and to the right along this continuum. You do this with consistent and thoughtful pressure, sometimes just by asking your subject-matter experts, team, and analysts if things have changed. Remembering to keep the focus on these items will help them come to a solution, one way or the other.

■ **Note** The goal with assumptions is to either retire them entirely or make them mitigable risks.

As mentioned earlier, one of the easy ways to uncover these things is to ask your team which decisions they feel least confident about. As this list grows, you should continually ask, “Do you still lack confidence in this decision? What would make you more confident? Do we have options if you turn out to be wrong? Is there anyone else we can get involved to help?” Continue to do this with each person on a consistent basis. Your attention and focus will help retire some of these issues—people will eventually become confident as they gain new information or will do the work to get the information needed in order to eliminate issues from the list. Other issues will remain uncertain, but your team may figure out contingency plans, moving them up into the upper-left quadrant. Or the team may know how to stop an issue from happening, thereby moving it to upper-right.

Merely tracking the risks and assumptions doesn’t help you. You need to actually change their state into something you can manage, mitigate, or influence. Although you’re still left with a handful of assumptions, they should only be a handful. And they should be the kind you would have no trouble chalking up to misfortune rather than poor or inattentive management.

What to Do When an Assumption Is Wrong

Sometimes, alas, you make an assumption that resides in the lower-left quadrant, and this assumption turns out to be wrong. There is nothing you can do to prevent it and nothing you can do to mitigate it. If you find yourself in this situation, then the first thing you need to do is be certain. You need to have a high degree of confidence that you’re faced with a lower-left problem. When you’re certain, you need to call it.

Your program manager won't like you coming to them with a problem without a recommended solution. But they will dislike you even more if you sit for too long on an important problem that you can't overcome. When you hit this point—there is a fatal flaw in your original thinking, and you can't stop it or work around it—you must inform your program manager. All you can do is recommend making a material change to the project, stopping the project entirely, or doing something else. Whatever the final decision, it's probably not yours to make as project manager; this is a portfolio-level decision.

The key is to remember that this kind of problem, although not terribly common, also isn't terribly rare. Your program manager has faced such issues before and should have some idea what to do. You should plan on being a participant and providing options and input, but you shouldn't try to solve the problem yourself. In fact, attempting to do so will probably hurt more than it helps. Remember this definition from earlier: "If we had known this from the start, we probably wouldn't have begun this project." If that is where you are in the process, don't charge ahead and hope for the best. That way lies madness, and you shall have none of that.

You never know: if you've kept your project in good order and have managed it well up to this point, then when or if the situation changes, you may get another crack at it. Or maybe you've moved on, and someone else will pick up where you left off and run with it (one of the ultimate tests of a good project manager). But if the decision is made to put the project aside, recognize that it's in everyone's best interest—most of all yours—to stop investing in something that can't work, so you can get started on something that might.

Summary

Assumptions are incredibly important not only to projects, but to everyday life. You make them all the time, usually without knowing it. Project managers, however, don't have the luxury of making assumptions casually. Assumptions need to be made on purpose, tracked, and monitored. Project managers need to move their projects forward, so they sometimes need to make decisions using imperfect, or no, information.

Identifying assumptions can be tricky, because sometimes they appear to be facts, or the person making them seems to be certain. One way to pick these out is through people's vocabulary: phrases like "I think" and "I'm pretty sure." If you hear this kind of statement from a subject-matter expert

or get a firm statement from someone who isn't an expert, you may be looking at something between a guess and an assumption.

Although it's important to track and monitor assumptions, you need to make sure you aren't tracking everything. An assumption like "There won't be a global economic meltdown" doesn't help anything. You need to track the assumptions that you can actually do something about and that will become risks to your project, not the world in general. Listing too many things only distracts you from the real issues you need to manage and monitor.

Once you have a list of issues, you need to continually test and validate them. Many of your assumptions will turn out to be true and can be crossed off your list. Some of them will remain open while you continue to test them, and others will turn from assumptions into risks. Risk management is its own discipline, but it's much clearer to manage them than to manage the lack of information.

When a project assumption turns out to be wrong, you need to judge whether it has a material impact on the project. If it does, you need to make sure the new information gets to the right people. The fact that an important assumption has turned out to be false isn't your fault, but not acting on it appropriately would be. Assumptions can be the foundation of a project, just as they are of your life; and when they change, you need to be ready to change your project to match.

Identify Decision Makers

“Whose call is this, anyway?”

Projects are a function of inertia, just like anything else. Steady forward motion, like that of a ball rolling down a hill, can be powerful and difficult to stop. This motion, however, is dependent on decisions being made in a way that makes sense to the people who have the authority and ability to make them, and not having those decisions overturned after you’ve based your project plan around them. In order to ensure this, the project manager must do a very deep analysis of a project’s stakeholders and identify who the key decisions makers are, what they want, and how to treat them. Managing this stakeholder group is one of the critical tasks of the project manager—and it does more to ensure success than nearly anything else.

Note that I mentioned two separate groups: project stakeholders and decision makers. Often, these people are the same. For example, the customer, sponsor, regulator, or implementer of a project is usually both a stakeholder and a decision maker. However, this isn’t always the case. You may find that the person who makes the ultimate decision about a project isn’t someone you think of as a client or a sponsor. You may also find that the primary beneficiary of your work has very little say in how the project is run or even what the project delivers.

This can cause an interesting problem for the project manager. You’re trained to understand the needs of the client; elicit requirements, constraints, and

goals; and then deliver a solution that maximizes value while minimizing cost, risk, and other factors. But doing so doesn't guarantee that your project is well received or viewed as a success. To achieve that, you need to grasp who the stakeholders are, who the decision makers are, how they relate, and how to manage them.

■ **Note** Stakeholders aren't always decision makers, and decision makers aren't always stakeholders.

As you move deeper into a project, your program manager assumes you've done enough diligence to determine who the end users are, what they want, and probably what their bosses want. More to the point, if you aren't doing this, then your program manager knows you aren't doing your job. But if you hope to do your job well, then you need to do a lot more than just meet the needs of your end users and their boss. You need to understand the entire ecosystem in which your project lives and determine what success looks like from the outside. Sometimes, success doesn't mean maximizing the value of the deliverable. Sometimes it's a combination of factors to which you need to pay attention.

As a project manager, it can be comforting to think only about the project you're working on. Figure out what needs doing, deliver as much as you can within time and budget, and move on. Sometimes this approach works. In fact, this is largely what consultants do: they figure out the maximum value they can provide and then do exactly that, without factoring in all the other considerations you need to worry about if you plan to work with this team or this client again. This is one of the many reasons consultants can get a bad reputation; they deliver what they promised, but it doesn't fit. It can be kind of like making a wish that a genie grants, and then discovering that although your specific wish was granted, everything around it was made worse.

If your goal is to manage a portfolio of projects and clients, then you need to learn how to understand that portfolio and how to deal with the multi-project give-and-take tradeoffs that can come with it. You need to understand when doing the right thing for the project may be the wrong thing for the portfolio or the relationships you hope to build within that portfolio. But before you're able to do this kind of analysis, you must figure out who the decision makers are and what they want out of the project. The closer those are to each other, the better, but you can't assume they're identical unless you do the research.

When it comes time to identify the stakeholders and decision makers, you need to think of the topics in the following sections and use them as a yardstick to figure out who you're servicing in this project. Without doing this analysis, you're left pleasing the wrong people, and you may damage your career.

Who Is Paying the Bill?

The first way to figure out who has the real authority when it comes to making the decisions for a project is to determine who is footing the bill for the work. Payment can come in many ways. It can consist of engineering hours, budget, hiring outside consultants, physical resources, or raw dollars. In a large organization, it can be as blatant as who signs the timecards. As a project manager, you should be able to figure out who is signing the checks that allow this project to exist. This is the person to start with when it comes to stakeholders.

Often, this person is in the management chain of the project's ultimate beneficiary. If you're building an internal tool, it's probably the boss or ultimate boss of the person or group that will use the end product. If you're building a product or a new sales tool, then it may be the marketing or sales department head. If you're working on a Human Resources Information System (HRIS) or finance tool, there's a good bet that it may be the head of HR or finance. This person should immediately reach the top of your list of people to talk with.

It's rare, but it does happen, that the person paying for the work either isn't an important stakeholder or has abdicated the role for one reason or another. Consider someone in sales management investing in a new commission-calculation application that resides in the finance department, or someone in customer service sponsoring a manufacturing change that will lower complaints. Or, imagine someone in a field office using their own resources to make a change to the way corporate does business, hundreds or thousands of miles away. In these cases, the people providing the funds that support the project either may not have the decision authority to decide how things are done or may feel they lack the skill to make the final decisions. This isn't necessarily a sign of a problem. In fact, it can be a good sign. The worst thing a project manager can have to deal with is a decision maker who lacks the skill to make the correct decisions but wants to make them anyway.

■ **Note** It may not be a problem that the person or group that is funding a project isn't the final decision maker.

Although this is a fairly unusual case, it's one you should be prepared to manage if required. More likely, the person who is ultimately funding the project will want the final say on key decisions.

The thing you must do is figure out who this person is. Depending on the size and cost of your project, and depending on the skill and seniority of the business owner, you may discover that the person who appears to be handling the bills actually has little authority. The task can be delegated, and managers often act as if they're the ones with final authority, but this isn't always the case. Sometimes the manager you're dealing with needs to go up the chain every month to get authority to keep the project going; this can happen if the project's needs exceed the manager's signature authority. It's also not unusual for a manager to be able to comfortably fund a project out of their own budget and not want the expenditure exposed higher or anywhere else. The truth of the matter is often not as obvious as you would like it to be.

How do you figure out who's paying the project's bills? If you're working with business owners who are acting as if they have final payment ability, how do you find out for certain? As usual, there is only one way. You need to ask questions like these:

- Is the funding for this project hitting your budget?
- Is anyone else chipping in?
- If we need to get more funding, can you approve it, or do I need to help you make the case to someone else?

These three questions tell you all you need to know, but it's important how you frame them. If you ask someone whether they have enough money to pay for the project, they will almost certainly say yes. Think about walking into a luxury car dealership: are you going to tell the salesperson that you don't have enough cash to buy one the show cars?

You need to ask your clients, subtly, if they're the ones signing the checks, if they're the *only* ones signing the checks, and whether they have any more to spend. And you need to ask in such way that the questions don't sound insulting. A car salesperson will ask something like, "Is anyone helping you out with the down payment?" Or, "How high can you go on your own?" These

questions are exactly the same thing—the salesperson wants you to be able to make the decision yourself but needs to prove that you’re able to do so.

Starting with the person who is providing funding is a very good idea. Most of the time, finding this person is fairly easy: work your way up the chain of the person from whom you’re gathering requirements. As with anything else, though, there can be complications. Maybe the person is getting help paying for the project, or perhaps they were granted a budget specifically for this purpose with no flexibility, or maybe they’re sharing the expense with someone else. Learning this person’s level of authority and autonomy is vital to determining the identity of the final decision makers.

Who Has the People?

Some say that it takes two things to make a project: money and people. You’ve looked at the money, and how the person who signs the checks often makes the final call. It can be hard to dispute the Golden Rule¹: “He who has the gold makes the rules.” But many environments have a different scarce resource: people. It’s not unusual for managers to have more budget than they have people to spend it on, causing a competition for resources that can frustrate and confuse business leaders. This is one of the reasons companies use consultants and outsourced development; there are business leaders with money to spend and nothing to spend it on. They often decide that it’s better to spend their budget inefficiently on an outside company than it is to not spend it at all. They’re wrong, but it’s hard to convince them of that.

But that’s not the situation here. You have a project team identified, and all of them know that they’re assigned to the project. That doesn’t eliminate their boss, manager, work leader, or resource owner from being a major decision maker in the project. Realize that if you’re in this type of environment, resource owners can be very selective about what their people work on. A manager may ask questions like, “What makes your project special?” “Why should my people work on your project instead of working with someone else who is willing to pay the same amount?” “What does my group gain from working on your idea rather than another one?”

¹ http://en.wikipedia.org/wiki/The_Golden_Rule#Alternate_22Golden_Rule.22_cited_in_business_and_politics

■ **Note** Resource ownership can mean money, people, or access to something. Each of these can carry its own weight and can rule the others.

Compounding the problem in a normal project, you probably have several such people to work with. Unless your project team consists entirely of people who work for the same manager, several managers will have those same prioritization questions. You need to convince each of them, consistently, that they're making the correct choice. Losing one or two project members can sometimes spell the end of a project, even if most or nearly all of the people are still on board.

The good news is that this type of decision maker rarely has the go/no-go decision in their hands. Nor do they have much say in prioritization or requirements gathering. However, they do make one important decision: whether their person continues to work on your project. This decision doesn't seem like a big deal; but if the team member is a rare or unique person, particularly a person who is in high demand from other projects, then the manager may decide to pull the person away and thus derail your project.

One again, there are three questions you should ask each manager of the people on your project team. If doing so isn't feasible, you can sometimes ask the team members themselves, so they can relate the information and stories back to their manager, should the topic ever arise. The questions are as follows:

- Do you believe in the business value of the project?
- How is this project helping the career paths or growth plan of the people on the team?
- Other than the business requirements, what else do you think this project should deliver?

Here you hit on all three inputs into the decision that the resource owner needs to make: belief in the project, belief in the side benefits of the project, and whether the project is good for the person's career or growth plan. A large percentage of the time, it takes a positive answer to only one of these questions to get the resource owner on board.

If you can get resource managers to believe in the value of the project on its face, then you'll almost always get their support. These folks have seen projects come and go, many of which they didn't believe in to begin with, and they turned out to be right. It takes a bit of selling and a spark of a good

idea to change their mind from thinking your project is just like all the others. If you have a good idea, then you should be touting it for all it's worth.

■ **Note** If your project seems like a good idea on its face, then you have an advantage in any prioritization discussion.

The second question addresses another problem that managers have on their minds: how to grow their teams. This can cover a wide array of topics, including cross-training, leadership growth, new skills acquisition, and providing an opportunity to someone who deserves it. Even if managers don't totally believe in the project, this can be a powerful motivator for them to support it. They have to live with the person for much longer than you do, and creating and delivering on opportunities can be one of the harder challenges that managers face. If your project can provide the solution to one of these problems, then the manager will probably favor it over others, simply because keeping people happy and engaged is a top priority.

Finally, if neither of these topics appeal to the manager, the last topic you can cover involves the other things you plan to do with the project. During the course of your project, you may plan to retire some legacy system that the manager no longer wants to support. Or maybe you're going to address an architectural weakness that has been plaguing the manager for years. Possibly you're going to fix a handful of bugs from the issue system during the course of your project, and that alone will make the manager happy.

But this is the weakest of the three motivators, for two reasons. First, these managers have played this game before, and they know that the improvements you plan to make and are willing to put in the project plan now are the first to get thrown overboard once the schedule gets short and resources wear thin. They won't be offended by the decision; they may even support it. But this means they know they can't count on it. Second, this kind of promise isn't unique to your project. Many projects include promises to make things better, improve weaknesses, or fix bugs. Although each of the promised improvements may be different, they're all desired. Because managers know not to count on such promises, all they need to know is that the promise exists.

Resource owners usually have one important decision to make: whether to let their people work, and continue to work, on this project. In order for them to make the decision in your favor, you need to either convince them

of the value of the project (unless you're lucky and the project's value is obvious) or solve one of their more difficult challenges: helping to grow their team. If you can do neither, then you're stuck with telling the manager what is in it for them and their team if the project launches. Be careful not to overpromise—the manager has had that happen before. But if you think you really can make things a little better, don't be afraid to play your card. Delivering on this promise will most likely put you in the top half of the resource owner's favorite project managers. You can use that later, too.

Who Is the Customer?

Few things can end a project earlier than having the primary beneficiary of the project declare that they don't want it. Having the customer decline to use whatever it is you're building—even before you're done building it—can kill a project immediately, and often it won't be considered again. You should never allow this to happen. For this discussion, let's consider the customer to be whoever is going to use the thing you're building, buy it, or use it to create value. This customer can be internal to the company or can be an end customer in the marketplace. It doesn't matter. A project that is suspected to be a failure in the marketplace will lose support, inertia, and, more likely than not, funding and resourcing.

The tricky thing is that customers may not be asserting that they don't want the outcome of your project. They may be saying they would rather have a solution to a totally different problem. This is akin to you saying that you don't want your housecleaner, because you really want someone to mow your lawn. If you're in charge of the housecleaning project, this can be a difficult problem to overcome, but you need to overcome it if you hope to have your project make it through delivery.

■ **Note** Sometimes, the client will claim to not want something, when what they really mean is that they want something else.

This is why aligning with the client, as described in Chapter 6, is so important. It can be easy for the customer to start asking for things you aren't prepared to deliver—and aren't even tasked with delivering. When you tell clients that they aren't getting any of those things, they will become more dissatisfied with what they *are* getting. This can be depressing when what you're planning to deliver is of provable value.

I once had a project manager working for me who was tasked with helping a company revamp its strategy for contacting customers. There was some light engineering support, but the primary focus was on using existing systems and changing the business strategy and workflows. About a month into the project, the project manager made a request for a substantial number of engineering hours, which amounted to about a 500% increase in the project. The client was willing to pay for it, so there wasn't a problem finding the people. The problem was that the project was heading in an odd direction.

Troubled, I asked the project manager what was going on. She told me that the company didn't want to address the contact strategy until it redid its e-mail, phone, and direct-mail systems. This would include vendor selection and a fairly large development task. The project manager was excited, because the project had suddenly become a much bigger engagement and involved doing some interesting work. However, this wasn't just a slight alteration in the original project—it was a material change in what we were expected to deliver.

This put us in a bind. The company did need a new campaign-management system, but doing that first was putting the cart before the horse. If we built a new way to touch customers *before* we determined what the company's plans were, then we were pretty likely to build the wrong thing. However, the company was certain that it needed new systems, new hardware, and probably a new department.

We wound up starting a new project to run in parallel with the original project. The original project manager was assigned to the new project, and I put someone on the first project who was less inclined to suggest a software build for a strategy engagement. If this had gone the other way, we could have built the company exactly what it was asking for, and it would have paid. But I'm convinced the company eventually would have blamed us for producing a really slick system when they had no real direction on how they should use it.

To avoid such problems, you need to do two tasks. First, determine if what you're delivering solves a valuable problem. You may not be dealing with the customer's biggest pain point or primary area of opportunity, but you need to confirm that what you're working on is of value. If you talk with the client and discover that what you're working on won't make much of a difference, then you may be better off stopping it and working on something else. But if you and the client agree that you're adding value, then you know you're on the right track.

Second, you need to understand what the client's biggest problem or opportunity is and whether its value is worth the risk of disrupting your project. Your client may be willing to say that your project, which is providing value, isn't needed, in an attempt to get a different project approved. That is, the client may be willing to play chicken with the process that approves projects, and risk getting no project at all, to try to solve their biggest problem. Sometimes this is the correct call. Often, it isn't.

Your task, as project manager, is to help customers understand their choices. It may be a question of delivering a solution for something that is a real problem and should be solved, or trading it for the solution behind door number one. There is no telling what will be back there when the curtain is raised, and you need to help the client do the math and make the decision.

The key, and what your program manager should drill into you, is the importance of making this decision intentionally. Clients can do damage to themselves by stopping a project that is in flight and getting nothing to replace it. If the calculations indicate that this is worth the risk, then you should be prepared for it to happen. But if the change doesn't seem worth it, or if the client doesn't realize the cost of what they're attempting, then you need to help them understand. Decisions made intentionally are usually correct, but decisions made with blatant disregard for the facts often aren't.

Who Has Signature Authority?

Sometimes, the person with final authority isn't the person paying for the work, the person who controls the people, or the end client. Depending on your project, final approval may reside in the finance department, compliance, legal, or the nebulous "brand" department. That is to say, the final signature authority on your project may be someone who otherwise has little stake in the project's success. Their job may be made easier if your project fails to deliver on what it promises.

This is one of the harder situations for a project manager to navigate. In the scenarios discussed so far, you could at least count on the notion that everyone involved wanted the same thing: to create value for the business. Once you start involving groups that don't have stake in the success of the project but are willing to limit your choices, you have an unfortunate conflict of interest to adjudicate. This can be especially difficult because the group you're referring to may be right, using some metrics, but they aren't using the metrics that the project team feels are the correct ones.

These groups tend to be *risk minimizers*, meaning their entire function is to approve only that which exposes the company to the least risk. Few companies succeed if they operate this way. The good news is that most people working in these kinds of groups realize that the business must take risks in order to succeed. The bad news is that risk mitigation is what they do for a living, all day, every day.

As soon as you determine that a group like this is taking interest in your project, you need to act. These groups, by nature, tend to be obstructionist; but worse than that, they tend to be slow. I've known compliance investigations to take several months and even make a project obsolete, because by the time they declared a product compliant, it was no longer needed in the market. If you wait, or allow these groups to operate at their own speed, then time will pass you by, and you might as well spend your time knitting.

There is a fine line you need to deal. If you proactively take your project to each of these groups and ask if they want to review it, the answer is more likely to be yes than no. That outcome is rarely positive for the project. But if you don't get clearance from these groups ahead of time, they can cause a sizable disruption when they decide to take an interest. What's a project manager to do?

■ **Note** Some groups exist merely to obstruct you. Deal with them carefully.

The answer is both simple and hard. You must build relationships within these groups, outside of the projects you're running. You need to find out if one of these groups is going to start paying attention to your project, but in a way that doesn't draw attention to your project. You don't want to fall prey to the *observer effect*—having such a group observe you change their level of interaction with the ultimate decisions. You, or your program manager, need to float the principles of the project through these groups as hypotheticals: if, say, you were to do this kind of thing, how much regulatory or compliance attention would be necessary?

This isn't an attempt to skirt legal or compliance review of your work—far from it. But you need to make sure that if your project is going to undergo a fairly intensive review, it's actually necessary. Realize that these groups exist to do these kinds of reviews, meaning their default mode is to hold up progress. If your project needs the review, then you should be prepared and ready to face it head on. If your project doesn't need it, then you don't want to spend resources on it. It's a careful balance.

Your job is to figure out whether one of these groups needs to be involved and, if so, when to involve them. Compliance (or any of these groups) can sink a project on its own. Your task is to not let such a group get involved unless they're needed, and if they're needed, to figure out how to comply with their demands. The worst scenario, and one that will get you little sympathy from your program manager, is having to comply with regulations that aren't particularly relevant to your project, simply because you failed to properly communicate with those groups.

Any Other Influencers?

Every stakeholder discussed so far has been fairly direct. These people can, through a single decision or signature, change the course of a project or cancel it completely. But these aren't the only stakeholders you need to worry about in the course of your project. Another class of people you have to consider aren't exactly stakeholders, in that they have little or no stake in the outcome, but are heavy influencers. These people, if they aren't on board with the idea or the project, can slowly and inexorably derail the entire effort.

These tend to be employees who have a long tenure with the company or were on a similar effort in the past. They can also be people who agree or disagree strongly with the idea. People who have been with the company for a long time sometimes feel like stakeholders simply because of their perceived ownership in the company—ownership gained through the time they have invested. And employees who have worked on an effort like yours in the past may either be rooting for it to succeed or rooting for it to fail, depending on how they feel about the idea.

Whatever the cause for their feelings, these people usually have the ability to create either a cheering section or a subtle infection for the company, project team, or project manager. Knowing who these people are gives you a great advantage when you're dealing with the general attitude toward your project.

■ **Note** Sometimes your greatest influencers come from outside your stakeholder community.

For instance, years ago, I was managing a program that was trying to get a social media project off the ground. It was approved several times, but each time it was stopped before it got started. Every time the project got ready

to start, the project manager and team—new each time—were full of energy and excitement for the new idea and new technology. But as it got closer, doubts among influential people started to creep in. Was there really a good reason to do this project? How important was social media to our strategy, anyway? Were the numbers used to make the case accurate, and was the value really there?

These comments never came from anyone on the project or in the project approval chain (obviously, because they kept approving the effort). They came from outside the project and outside the group, from a host of people who didn't want the project run, or wanted us to work on something else, or wanted to be the ones who were on the project instead. Inevitably, enough negative comments came from the gallery that project momentum slowed, facts needed to be checked and double-checked, and the project was delayed while the company "reassessed the opportunity."

After the third year in a row of having this happen, the primary business owner gave up on the idea and moved on to other areas of the company. At the start of year four, the project-prioritization committee asked, incredibly, "Where is the proposal for the social media project?" A new business owner was identified, a new business case was created, the project was once again approved—and a few months later, it was unceremoniously cancelled. I could probably have predicted it.

Does Anyone Want This to Fail?

And that brings the discussion to the last kind of project decision makers you have to work with: people who want the project to fail, for one reason or another. These people can exist at any level, from team member to executive, from a sponsor of the project to someone in a different division. Few of them will come out and tell you this, which makes the prospect of discovering them all that much harder.

For instance, the VP of HR may want the new HRIS project to fail because it means she will have to lay off 20% of her staff, even though she is sponsoring the project. Or someone in sales may want a new project initiative to fail because he wants the company to focus on improving the quality of existing products. Maybe another project manager, who failed in delivering this kind of project in the past, doesn't want someone else to succeed. Or perhaps someone doesn't believe the project is worth doing.

Whatever the reasons, from jealousy through good citizenship, you may find that key influencers want your effort not to succeed. As usual, the way to

find these types of people is to ask and interview. When you're interviewing or chatting about the project with team members and stakeholders, ask them, "What are your concerns if this project succeeds?" There are two important unstated inferences in this question:

- *You're assuming the project will succeed.* This is specifically the tone you want to set, not only within your project team, but with everyone. They need to envision the project launching, in order for them to get their fears out in the open.
- *You're assuming they do have concerns.* You aren't asking whether they have concerns; you're asking what those concerns are. This gives them permission to have these concerns and a way to talk about them in a safe environment.

Finally, at the end of every discussion like this, you should ask, "Do you know anyone else who would be concerned with this?" The result will be honesty. People are usually willing to conjecture about what other people are thinking. You get statements like, "I don't know if she's worried about it, but this system is going to take the place of ten HR reps. I wonder what the VP of HR is going to do about it?" Or, "When we tried this last time, one of the architects was adamant that it couldn't be done. I don't remember his reasons, only that he seemed certain." These are the people you should talk with next, to ask them about the concerns you're giving them permission to have. Sometimes, just allowing them to voice their complaints is enough to win them to your side.

Conversations like these almost always have a positive ROI. Giving up 30 minutes or an hour can win you back months of project time, either by preventing a problem or by getting a key person aligned. This is exactly what you need.

How to Manage Stakeholders

When you're dealing with this disparate group of stakeholders, you need to find out where they lie on the influence/interest graph shown in Figure 8-1. Influence can be measured in many ways, from having budgetary and resource control all the way to being one of the popular engineers at the lunch table. All of these people have the ability to set your project on a different trajectory, sometimes without you around to do anything about it. Determining where they fall on the matrix presented in Figure 8-1 helps you figure out how to deal with them.

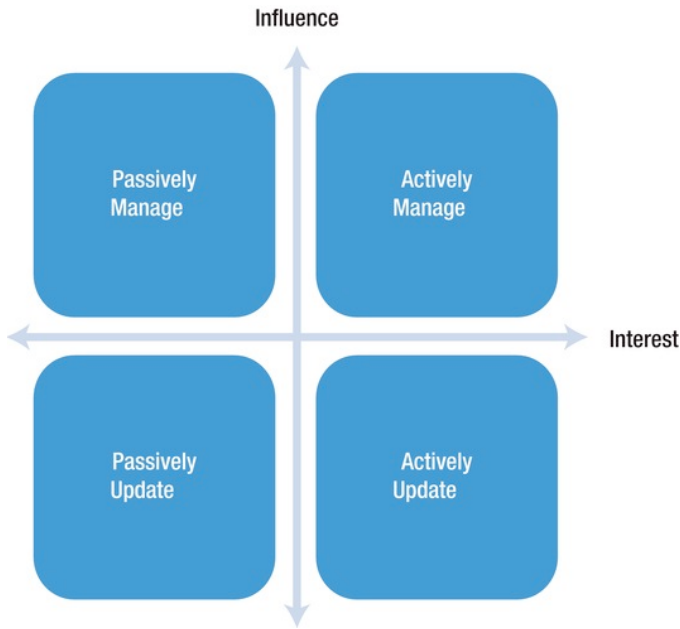


Figure 8-1. The influence/interest matrix

Some people have a high degree of influence but don't have a lot of interest in your project, or they're unwilling to spend their influence on your project. These people need to be passively managed. It's important that they maintain a positive opinion of the project, but it's not important that they stay informed about the project's details or progress. However, you need them to keep thinking the project is running fine, has a competent team, and is a reasonably good idea. You should provide these folks with roadshow-type communication: on an infrequent basis, you hit the highlights of what is going on, what is complete, and what is yet to come. Whether your project is ahead of schedule, over budget, or at risk of total collapse has little bearing to these people. Give them a highlight reel once a month or less, and let them go on thinking about other things.

Other people have a low amount of influence and a low amount of interest in your project. They may not be thinking about your project at all. However, they may start paying attention if something interesting or disastrous happens with the project. These people should receive little or no communication from you after your initial discussion with them. They should know where your project portal is, how to get into your wiki and find your documents, and your email address. If they want more information about the

project, they should either search it out or ask you. You don't have enough time in the day to keep everyone up to date, and this group will have to do some homework if they want to be in that group. Careful, though: if people you've identified as being in this quadrant begin seeking out information, they may be moving into a different classification.

The lower-right quadrant of Figure 8-1 contains an important set of people for your project. They care quite a bit about the project but have no authority or ability to change it. This group is often made up of individuals who have either thought about the problem a lot or been brainstorming solutions or alternate approaches on their own. They can be a huge asset to you and your team, because they can sometimes be used as free resources—that is, their level of interest may be so high that you may be able to get work (even if it's just a single thought or idea) out of them without having to put them on your project. This is doubly in your favor, because good ideas can come from someone who has other responsibilities but is motivated enough to send their thoughts to you.

You should actively update this group. Keep them on your regular communications list, offer weekly updates, and let them know whatever is going on. Spare some time to chat with them every now and again, especially if you have a problem or challenge you think they may be able to help with. These people *want* to help you, so do what you can to give them the information they need. If you do this well, you turn them into active supporters, which can be valuable if the project needs them in the future.

Finally, the upper-right quadrant is made up of people who pay a lot of attention to the project and have the authority to materially impact its future. These people may be executives, division heads, managers, or anyone mentioned earlier as having the power to create change. This is your primary customer group, and you need to keep them happy throughout the length of your project. The worst scenario for you is if they don't feel they know what is going on; and it can be difficult for you to control what they feel.

Keeping this group happy doesn't mean just giving them the good news or hiding problems or issues from them. They have something invested in the outcome of this project, such as time, money, resources, or success. In order to be satisfied, they need to believe that the project is being managed by a competent manager, and that they know enough about what is going on that they can see it for themselves.

Note Many executives are watching you run your project but aren't focused on the project itself.

This group should get a treatment that is a combination of the other three groups. Of course, they should get the location and access information for your project portal and documentation. And you should share project successes with them. But you also need to actively meet with them to find out what they're thinking, give them the current status, both good and bad, and make them feel like part of the team. The value of this group can't be overestimated. Not only can they stop a project with a single e-mail, they can also do the reverse and add resources, budget, or time. They can remove obstacles or make a key introduction. In short, a well-cultivated relationship in this group can be a project benefactor or even fairy godmother.

Neglecting to provide this group with the attention it needs can be a double negative for your project. A bunch of influencers will be seeking information about the project on their own, and you may miss out on the kind of assistance that only someone in this quadrant can provide. One of those two reasons is enough to make the time to communicate, but having the potential for both makes it a must do.

Summary

Identifying your project's stakeholder community is vital to the project's success. This community includes people with heavy influence on the project, key gatekeepers of money and resources, people who have a lot of interest in the effort, and people who have a lot riding on the outcome. There are also some negative influencers to manage: those who want to stop the project in order to fund something else, or even want the project to fail.

Knowing who these people are, and knowing their level of power and energy for the project, is important for knowing how to best manage them—and whether you should manage them at all. Not finding your stakeholders and decision makers means they have to find you. That is a situation you should strive to avoid.

Communicate Effectively

“I must not have gotten that e-mail...”

Virtually every project post-mortem starts with the same item in the Do Better Next Time column. The session facilitator dutifully writes **Communicate Better** on the white board, turns to the team, and says, “What else?” Although “Communicate Better” is written in dry-erase marker, it might as well be written in Sharpie, because it seems to be a permanent member of the project wrap-up process.

This always wounds the project manager. They’re thinking, “We had weekly meetings, a status e-mail every Friday, monthly updates, and a quarterly executive briefing. We even had a few daily meetings during construction. I felt like all I was doing was communicating!” The project manager is probably correct—they were doing an awful lot of communicating. But to be fair, the software engineers were also doing a lot of coding, and the analysts were doing a lot of analysis. As with many other things in life and work, the importance of the quantity of delivery pales in comparison to the quality.

■ **Note** Communicating better doesn’t necessarily mean communicating more.

Your team isn’t asking you for more communication. In fact, they may be asking you for less of it. But what you do provide needs to be higher quality,

be in a different format, include different content, or differ in some other way. As a project manager, you need to keep two things in mind, each of which differentiates your role from the members of your team.

First, although you may be very clear about what the project is attempting to do, the goals, the current risks, and where you are in the plan, not everyone on the team has this knowledge. Sure, you may have wonderfully concise, pithy, and direct project principles, and you may have printed the project charter on posters and hung them around the meeting rooms. But no one on the team is living the project like you are, and even if they're hearing the information, few are internalizing it. That's your job, not theirs.

Second, information sharing isn't the same as communicating. In meetings, e-mails, reports, and updates, you may feel fortunate if people are paying attention half the time and reading everything a quarter of the time. Any communication that is meant for more than one person almost certainly contains information that isn't relevant to any particular reader. Your team members know this, and they have built reading habits because of it. Even formatting certain things in bold red doesn't help, because although those lines catch the eye, the first time someone reads a highlight and it isn't relevant to them, they begin to distrust your highlights.

In today's environment, one of the key areas of focus is return on investment (ROI). This used to be reserved for decisions related to budget, capital expense, or resourcing hours. Now you calculate ROI on nearly everything, including your attention and personal time. If an activity such as reading an e-mail proves to have too low a return (value), then you limit your investment (time spent) on that activity until you reach the proper ratio. And once your communications dip below the positive line, it can be very difficult to get anyone to listen to what you have to say, no matter how important it is.

Your task as a project manager is to deliver information and foster good communication across multiple team members, multiple functions, and perhaps even multiple locations, geographies, or time zones. Technology has handed you several new communication channels, and you need to use them appropriately and wisely in conjunction with the old and archaic ones. Finally, you must make certain your messages are worth reading and, even more important, worth understanding. Without proper communication, a project is very unlikely to succeed as you intend.

A program manager doesn't want to hear this from a project manager: "I sent him an e-mail...." It shows the program manager that you didn't do your job. E-mail, reports, and updates in themselves aren't communication—

they're communication vehicles. In project management, the medium is *not* the message. In your case, the message is the message.

The Importance of Communication

Projects are coordinated efforts performed by several people, all with the same goal in mind. That may be a simplistic definition, but it's as good as any other. It's possible for disparate groups of people to all work toward the same goal, doing incremental tasks that eventually reach the correct result, without active coordination and management. Lots of important advances in civilization were made this way. No one can claim to have been the project manager for project "creation of fire" or "use of the wheel" or even "ubiquity of e-commerce." But these efforts took decades or centuries to deliver and propagate. If you have deadlines that measure in months (or weeks!), this is probably not the best strategy.

When enough smart people are aligned on a problem and can work together, build on each other's successes, and understand their role in the solution, magic can happen. You're tasked with creating this magic over and over again, often out of thin air. And it's primarily your skill in communication that allows you to make it happen. A project can survive a subpar design, execution that is only adequate, and even a below-average team. But few projects can succeed with poor communication and the misalignment that usually results.

Chapter 6 talked about project alignment and the fact that no matter how slick your vehicle, if one wheel is misaligned, it creates significant drag on your progress. If more than one of your wheels is pointing in the wrong direction, you can probably forget about getting wherever it is you want to go, plus you'll have a pretty bumpy ride along the way. Effective communication is the primary instrument you use to create this alignment.

As much as this seems like a tautology, it's disheartening to note how often communication shows up on the "do better next time" list. If the team recognizes the importance of it, and everyone makes serious efforts, then why is it still a problem? The answer is both simple and complex. What is effective to one person isn't effective to another; you may need to reach several different audiences, each with a different preferred method of getting information. Or perhaps the specific skills that you have as a project manager don't suit the audience you need to address. For example, you may be fabulous at creating and giving presentations, but your team is so geographically diverse that there is no forum to do so. And finally, there is no single correct answer. What worked last time may or may not work this time. You won't know until you try.

The struggle is that talking about communication isn't enough. Think of the situation a baseball pitcher faces. Each batter is a new and different challenge and requires a new strategy. A pitcher who has only one pitch can't vary his style, and what may have gotten the previous batter out won't necessarily work this time. Not recognizing that each batter is different, and treating them accordingly, leads to mistakes or a subpar performance.

Effective project communication is similar. You must recognize that each interaction can be different and require a different set of skills to properly execute. To do that, you need to excel at more than one communication style and be able to match your approach to the situation. However, unlike in the baseball example, where each interaction is purely one on one, your communication tasks take on a wide variety of forms, from individual, to group, to many communicating to many. So you need several ways to communicate with several different types of people and to a varying audience size. It's no wonder that effective communication shows up in so many project post-mortems; it's a complicated and difficult thing to get right.

Differentiating Communication for Different Audiences

For most projects, you communicate with at least four different audiences, and each has a different set of needs and challenges:

- The internal team
- The extended team
- Other departments or functions
- Executives and customers

Let's look at the communication needs of each group.

Communicating with the Internal Team

The *internal team* is made up of the people who are working on the project and have at least some responsibility in making it a success. I often refer to this team as the people who self-identify as being on the project. They aren't merely contributors—they're full-fledged members of the team. This group has the most communication needs simply because they're managed the most. It's also the team for whom the content of the message is more important than the message's style, medium, or format. Your team needs clarity and consistent realignment on the following:

- *Roles, responsibilities, and goals:* Team members need to be clear on what their roles are, what they're responsible for, and what the goals of the project are. This is even more true if any of these things change. Over the course of a project, you may well need one of your team members to switch a role, or one of the goals of the project may be altered. But even if nothing changes, a constant reminder of what everyone should be doing is vital.
- *Task coordination:* Your team is working on tasks, both large and small, as the project moves forward. If you have more than one project team member, then their work requires some degree of coordination. Often, the team can be counted on to manage this kind of collaboration themselves; but just as often, a little help from the project manager can go a long way. After all, you want your team focused on delivering, not necessarily on coordinating delivery with others.
- *Project updates:* This is frequently the most bland of the communication needs, but it's important. Everyone on the project needs to know how the project is tracking to the plan, what the risks and challenges are, and what the successes have been to date. It's important to send this kind of communication on a regular basis. If you need to tell your team something impactful, you want a standard vehicle to do so. A project manager who doesn't send out regular updates raises eyebrows when they send information to the team; it seems more out of place than it probably should.
- *Decisions and changes:* During the life of the project, certain key decisions are made and unmade, the project may alter course, or actions may be blocked awaiting some action. Sometimes, your team is awaiting a go-ahead signal or permission to proceed with something. It's important that all decisions are well known to the entire team and that people aren't waiting for an answer that has already been made.

Communicating with the internal team happens in several ways: e-mail, meetings, instant messaging, telephone, and simply walking around and talking to people if you're co-located. It isn't uncommon to have interactions with everyone on your team several times a week. Some of this happens on an as-needed basis, and some of it comes on a regular schedule. The format and frequency is less important than getting the information out.

■ **Note** With your internal team, making sure the message content is received quickly and accurately is more important than the format or style you use to convey it.

Communicating with the Extended Team

The *extended team* is made up of the people who are working on the project but don't consider themselves team members. These are mostly people who have a limited role, are providing a key service, or are otherwise contributing but have little or no real responsibility for the success of the project. It can be someone as vital as the IT guy who is setting up your project server or an artist who is providing a few graphics for the launch.

This group is similar to the internal team, but its information needs are somewhat different:

- *Requirements, deadlines, and deliverables:* These types of people are often on multiple projects at once. Because of that, they don't have the personal bandwidth to carry a deep understanding of your project in their head at all times. People in this group need to know what they're supposed to be doing, how they're supposed to be doing it, and by what date you need it. Your project is but one of many, so they're likely to prioritize based on the nearest deadline, not necessarily on which project is the most important. The worst thing that can happen is that someone ready to work on your request lacks the information to do it or finds it unclear. This will almost certainly result in your task being moved to the back of the line, being delivered incorrectly, or arriving past the time you need it. Clarity of communication is the key.
- *The value of the work:* Most people on your project team know why they're doing whatever task you have them doing. All the items on their list are worth completing, and they tend to understand the value of tasks that seem unrelated or tangential to the project because they have a good grasp of the greater goals. This is rarely true of your extended team. Often, all they have is a window into your request, without context or knowledge of why it's important to your project. With your internal team, telling them what you want done is usually enough to get them started. With your external team, the "what" needs to be accompanied by the "why," so they understand the value of what they're doing.

- *Project status:* This is different than the project status you communicate to the internal team. In this case, the extended team needs to be reassured that your project is still running, that your requests are still needed, and that someone is waiting on them to complete it. Conversely, if none of these are true, if your project is cancelled or on hold, or if the task has sunk lower in priorities, they need to know this as well. You don't want extended team members servicing requests that are no longer needed. Although it may not hurt you right now, it lowers your credibility for the next time.

Communications with the extended team most often occur via non-real-time methods, such as e-mail, reports, or a request system like a ticketing system. This is usually because when the project manager has information to impart, recipients aren't ready to receive it. They may be working on something else, or they won't be handling the request for weeks or months in the future. They care and they want the information, they just don't want it right now. And the more you press them to make sure they understand, the more time you take away from someone else.

Communicating with Other Departments and Functions

The largest group you need to communicate with are *other departments or functions*. These people need to be notified about what you're working on, when it will be delivered, and what it means to them. However, they have very little stake in the outcome, if any. Many of the people in this group have no context about the goals of the project and why it's important. Nor do they necessarily have the time and inclination to learn. Even so, your project will likely run better if you include them in communications.

■ **Note** Even groups that have very little to do with your project need to be communicated with.

This group may contain people who happen to have knowledge you need, contacts you can use, or an opinion you value. They need to know exactly two things:

- *The problem you're trying to solve:* Your project is trying to solve a problem or expand an opportunity, or possibly both. Other departments need to know what you're attempting to achieve for two

reasons. First, you find out whether anyone else has already attempted it and has insight or information you can use. Second, you may learn that the problem is more widespread than you realized, that there are requirements you're missing, or that resources are available to you if people know what you're doing. There is usually no downside¹ to letting enough people know what your goals are and seeing if they're willing to help.

- *Roughly how, and about when, you plan to solve it:* This doesn't mean you need to share your solution design or master project schedule with everyone. Normally, taking a guess at what quarter the project will launch is precise enough. And giving a hint about how you're attempting to solve the problem, even at a very high level, gives the rest of the organization some level of understanding of why it will take as long as you propose. If you're solving a simple problem with a simple solution, but you publish a four-year timeline, people will ask questions. Similarly, if you're solving a hard problem in a difficult manner, but you plan to be done by next week, you'll probably get some sarcastic remarks.

People in different divisions or departments should know as much about your project as fits on a 3 × 5 inch card. Include what the problem is, 25 words or less about the solution, and roughly when they can expect to see it. This is enough to get the information across to those who aren't interested but need to know and to spark the interest of those who have a stake or something to add to the project team.

Communicating with Executives and Clients

Originally, I planned to list communicating with *executive or clients* as two separate audiences. A deeper analysis shows, however, that they have similar needs and require similar treatment. If you look under the surface, executives and clients tend to share a common characteristic: they're usually the ones paying for the work. They want to know what is going on, and they want to have a high degree of confidence that things are proceeding as they should be, but they don't want to manage the project—that's why they hired you in the first place. What makes this group slightly different is that your

¹ Originally this sentence was missing the "usually," but we've all had projects that needed to be kept secret for one reason or another. If that is the case with your project, you probably already know about it and control your communication accordingly.

communication with them should be equal parts what they need to know and what you need them to know. Sometimes, those are different things.

Three classes of information are relevant for this group:

- *Expectations:* What can the client or executive expect from the project, and when can they expect it? Those are the two questions this group is always thinking about. It's your job as project manager to manage both sets of information. They may be getting features, functionality, savings, opportunity, or a host of things. When they will get it can be a flexible answer. A project goes through several stages, from feature complete, to ready for testing, to a beta launch, to a worldwide external launch, plus points in between. Make sure your project has enough time to deliver on its commitments while putting something in your clients' hands as soon as possible. Often, the "what" side of the equation doesn't move much during the project, but pay attention to your definition of "when." The more skilled you are at positioning the various types of release dates, the more you can simultaneously show progress and continue to work on your solution.
- *Status, risk, and progress updates:* This isn't much different than other groups, other than the fact that you're talking at a higher level. Executives and clients don't care much for internal milestones, particularly ones that seem made up, such as "50% complete." They want to know if the project is on track and what risks exist that may change that. If your project isn't on track, you should be upfront and honest and not attempt to omit or obfuscate bad news. But if your project is doing fine, more or less, then you should feel comfortable saying that, too.
- *What they can do to help:* Any good executive will end the discussion with this question: "What can I do to help?" You must go into the interaction knowing your answer. Often, the answer is "Nothing." The project is proceeding as planned, and there's nothing they can do to help at this moment. But sometimes, you can make requests that greatly increase your chances of success. Don't hesitate to make such a request, but plan it ahead of time. Failure to do so leaves you unprepared in your attempt to ask for help—which doesn't instill confidence.

■ **Note** Sometimes, being prepared with your answer is as important as the answer itself.

Communicating with your client or executive team can be high risk / high reward. If you aren't successful in properly setting expectations, you can wind up with disappointment at the end of the project. If you aren't prepared with requests for help, then your opportunity to get assistance may pass you by. And finally, they're expecting you to manage the project. If they sense that you're struggling, or that the project would benefit from a change in management, they probably won't hesitate to make a change. Using the communication techniques outlined here is how you convince them that things are going well and a change isn't necessary.

Each group you're dealing with has a different set of communication need. Figure 9-1 diagrams these concerns. No wonder it's impossible to send a worthwhile e-mail to the entire list of people involved in the project—no matter how complete, most of it won't be relevant to someone.

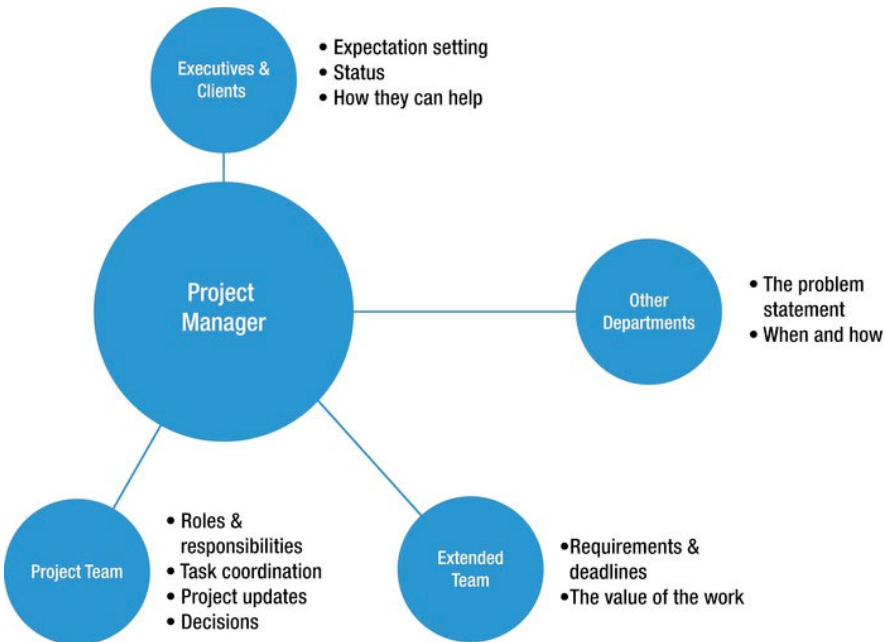


Figure 9-1. A communication diagram

Challenges of Different Audiences

In addition to having different needs, each group also comes with its own set of challenges. Not only does each group need different content in their messages, but you also need to be mindful of how you're delivering those messages. You must also be aware of certain pitfalls that may be in your way.

For instance, let's talk about your project team. This is the team you work with most often. You probably know most or all of the members, and you feel comfortable talking with them through many different mechanisms—and you do so often. However, with that trust comes a certain level of responsibility. For example, imagine a fairly benign statement like, "Let's be more careful this time." If it came from someone outside of the team (say, an executive or client), then it would seemingly be directed at the entire team. But coming from the project manager, it can be taken as a personal attack on the person who wasn't careful last time. You, the project manager, know who that person is; and an apparently tame, impersonal statement may be construed as calling someone out specifically. Be careful not to damage relationships you've built due to sloppy communication.

Or consider your extended team. They may have built-in skepticism and ask for evidence with regard to value and deadlines. Everyone thinks their request is the most important and needs to be done as soon as possible; you're not alone in this situation. So, when you make repeated requests, people stop believing your assessment of their urgency; and the more you stress their importance, or yell, or stamp your feet, the more likely they are to feel correct in their distrust. With this crowd, you need to be rational, understanding, and accommodating, while applying gentle pressure and providing information that will get your requests completed.

■ **Note** Not everyone involved in your project will be inclined to believe you. You need to manage this in your communications as well.

Another challenge comes when you're speaking with your project's executives, sponsors, or clients. Chances are, they have been lied to in the past; and more often than not, they're suspicious of good news or anything that feels overly aggressive. They want to know what you aren't telling them, and they will poke at most of your statements to try to ascertain their veracity. If you plan to make an assertion that everything is going well in the project,

then you not only need to be sure it's the truth, but also have enough evidence and information to provide a level of comfort that you're correct. When you speak with your project team, they're inclined to believe you. When you speak with your client, they're inclined to believe the evidence, so you need to have it in order to win the day.

When you're dealing with multiple audiences, you must take into account the information they need as well as the challenges you need to overcome to get them that information. These challenges should inform what you say and also how you say it and how much care you take during your preparation. Treating all groups as if they have the same needs and the same obstacles can lead to communication that is ineffectual or damaging to your project.

Treat Communication as a Project Deliverable

For a considerable amount of time during a project, the main thing you provide your clients and stakeholders is communication. Reports, updates, prototypes, decisions, documents, and so on form the bulk of what you're producing. This is even true using iterative methodologies. Until you have something that is working enough to launch into the market, all you have is your word and your skill to provide to your stakeholder community. For many projects, your communications are all they receive for at least half of the project duration. With that in mind, you need to treat your communications as a project deliverable—and one of the more important ones.

Consider your communication efforts as similar to the product or project you're building. You should gather requirements, constraints, needs, and expectations. You should test your assumptions, gather risks, and produce a communications plan and schedule—and stick to it. The less work you do in this area of the project, the more likely you are to get your communications wrong in content, form, or frequency.

One of the worst things you can hear about your project is that people who feel they should be informed aren't. It's even worse when the person who claims to not know what is going on is on your monthly (or weekly) update list, gets all the documents and e-mails, and has the same access to the project wiki page or portal that everyone else does. You may be bewildered that people can claim to be uninformed when everyone else seems satisfied with the level of interaction they're getting from you.

Chances are, if you're in this situation, it's because you never asked those people what they wanted—you never gathered their requirements. You

don't know if they prefer a PowerPoint presentation to an e-mail, or if they know how to check the wiki. Maybe they need a ten-minute phone call once a month, or maybe what they really want is for you to keep someone on *their* staff informed. The key is that you probably don't know because you never asked.

■ **Note** How your stakeholder community wants to be communicated with is a key piece of information that you must have if you hope to be effective.

Look at it this way. Not ensuring that you have all the requirements of your stakeholder base collected, prioritized, and analyzed would seem to be a failure in project management. If a key stakeholder says that production must be done in the United States, and you don't know it or haven't determined the impact that will have on the project, then you haven't done your job correctly. It's the same when it comes to communication. Talking with every key stakeholder, and finding out what they want to know about the project, how they want to learn it, and how often they want to keep in touch is must-have knowledge. Without knowing what their desires are, you can't make an intelligent choice about your communications plan.

But even this isn't enough. Assuming you know how your stakeholders want to be communicated with, and you've taken pains to reach a compromise or discovered a way to tailor your style to their needs, you still need to know how you're doing. Just as you send software through QA and test the first few widgets that roll off the production line for adherence to specification, you must test your communications. And the way to do that is by continuing to ask. Every so often, ask the stakeholders if they're getting the information they think they want, or if they feel informed about the project. The act of simply checking will almost certainly be illuminating. You may find that your messages haven't been read or understood. Maybe there was an organizational change of which you were unaware, and you're now talking with the wrong people. Or perhaps there are opportunities for you to do better or to take advantage of something else that is going on to make your task easier or make your communications more effective.

Treating communications as a deliverable should emphasize in your mind how important they are. You need to know what people expect to receive, how, and when, and you should constantly test and improve your communications. For much of the project, communication is all you provide, and not

knowing what you're supposed to do and keeping track of how you're performing means you're executing blindly on a key part of your project.

Communication Channels

So far, this chapter has talked a lot about what to communicate and how to tailor not only the message but the style of communication depending on the audience. Now, let's cover some of the important communication channels that are available to you as a project manager and some of the strengths and weaknesses of each. Often, you're expected to work with lots of different people over several different channels, sometimes simultaneously. Let's break these channels into three groups: broadcast, individual, and closed circuit.

Broadcast Communications

The first type of communication, *broadcast*, is the one project managers use most frequently. Consider the way broadcast television or radio works: content is created entirely out of sight of the audience and then revealed to everyone at the same time without affording them the ability to ask questions or change things along the way. This kind of communication usually takes the form of update e-mails; reports; updates to the project's shared area such as SharePoint, a wiki, or a project portal; and documents such as requirements and issue lists. Project managers generally work on these on their own and then publish them broadly to the entire project team or community.

■ **Note** Don't assume that all your broadcasts are read by everyone; in fact, assume the opposite.

Broadcast communication is generally best when you want to send a lot of information to a very large crowd, but you don't have the time or inclination to accept input from that same crowd before you publish. This could be because the information isn't all that impactful, such as a regular status e-mail; or it could be because the group you're sending it to needs to be notified about the information but is powerless to change it. As a project manager, you should never expect that your broadcasts are seen by everyone, just as a TV station shouldn't assume that everyone watches all of its programming. But it's important for your project team to know when to expect these communications from you and where to find past "episodes" for reference.

Individual Communications

Individual communication can take several different forms. It can be a one-on-one meeting, a phone call, an e-mail, an instant message, or a hand-written note. The purpose is to tell a specific person something and make sure they understand it. These communications run the gamut from true real-time, such as a face-to-face meeting; to real-time or near real-time, such as an instant message or phone call; to delayed reaction in an e-mail or a note. Whatever the case, you're asking for a response, even if that response is "message received."

You use individual communication to assign tasks, answer questions, thank or appreciate someone, or ask what people want for lunch. This kind of communication is how relationships are built and teams are forged. The ability to connect with your team on an individual level greatly improves your ability to manage them, both in this project and in future projects. In large teams, it can be tempting to not attempt this kind of communication and only talk with leaders who then relay your message to others. Depending on the size and scope of your team, this may be okay; but individualized messages, even if they come very infrequently, are valued by nearly every recipient.

Closed-Circuit Communication

Finally, there is the kind of channel I refer to as *closed-circuit* communication. This kind of structure involves more than two people, but you must be present at the time in order to participate. Such communications include meetings, conference calls, large teleconferences, and chat rooms. The venue is intended to be interactive like individual communication but also to have some of the best parts of broadcast communication. For many meetings, the facilitator can prepare the information ahead of time in the form of a presentation or other meeting materials, but still seek feedback and field questions in real time. Additionally, the participants can see and hear each other, and can react and build on statements made by others.

This kind of channel is best used for two things: fostering alignment and driving to a decision. It's difficult to align a team one member at a time. Team members like to ask questions and register objections in a way that everyone can see and discuss. They also like to see other people agreeing, because it makes it easier for them to agree, as well. Alternately, you may have a difficult decision that needs to be made, or you need to escalate something to allow for discussion. One of the best ways to do this is to gather the key decision makers together and have them make a decision. As soon as the number of

people you need to align is greater than two, attempting to do it any other way takes significantly more time.

Understanding the content, style, and channel for your communications is important in making them effective. Each group has different needs, and each comes with its own challenges. Selecting what form your communications take is often as important as what you say and how you say it. Make this choice intentionally.

Determining Whether You're Being Effective

Even if you feel you're recognizing your audience, tailoring your message and medium to the group, and getting the responses you want, you need a way to determine if you're truly being effective. As you probably can guess by now, the best way to do this is by asking. You want to know three things from your audiences: are they getting the content they need, are they getting it in the form they want, and are they getting it with the frequency they desire? Actually, there is a fourth question: do they still want to receive anything from you at all?

The best time to ask this question is right after a communication event. Asking this kind of question after sending a weekly status update, or after a meeting or a phone call, is a perfect time to get feedback on the level of satisfaction your recipient has with your communication. This is a one-on-one type conversation. Many project managers include a line such as, "If you no longer wish to receive these e-mails, or know someone who should, please contact me." However, the person who needs to act on that request probably isn't reading your e-mails to begin with.

Trying to poll everyone in your project community may not be feasible, and you shouldn't try. But without asking your key constituents, there is no way for you to learn whether your communications are valuable, helpful, and going to the right people. And without knowing that, your communications are mostly worthless.

Several years ago, one of my project managers was kicking off a project that we had tried to get off the ground before. We were going to scrape executive information directly off company web sites, as a way to have the best and most accurate executive listings. We had also failed at this before, but old ideas die hard.

I knew that one of the VPs from back then, a guy named Brad McCain, had a lot of thoughts about the project. He had moved on to a different division, but he was a pretty smart guy, and I remember he felt strongly about the idea, so I suggested that my program manager reach out to him. A few weeks later, I asked her if Brad had anything interesting to say, and she replied that she sent him an e-mail and added him to the status-update list, but she hadn't heard back. Not a big deal, people are busy, or maybe he no longer cared.

A few more weeks passed, and I happened to bump into Brad at lunch. I asked him his thoughts about us spinning that old project back up. Of all the responses I was expecting, the one I never considered was, "I had no idea. When did it start?" After filling him in a little, I went back to my project manager and asked about it. Turns out she had contacted a different VP, a guy name Brad McCarty. That Brad had no interest in or knowledge of the project and was filing away her e-mails unread. She was sending information to the wrong person, and she never knew it!

Other Benefits of Effective Communication

Communication is a core skill of project managers, and it's the primary way you can manage your projects, direct your teams, and satisfy your clients. You need to be able to communicate effectively in order to create alignment around project goals and requirements. Without that alignment, you can't hope to deliver. But the advantages of effective communication include much more than that: fostering team growth, better use of communication avenues, better expectations for your clients, and improved visibility for the project manager.

Only two questions matter at the end of a project: does the client want to work with you again, and does the team want to work with you again? As many project managers know, working with the same team on multiple projects is a huge advantage.

The best way to grow this team is through effective communication. You want the people on your team to feel that they know how to get the information they need, when to expect it, and how and when to voice whatever was on their mind. This helps create a team that is more engaged with the

project and with you as project manager. It also fosters the desire to work with you again.

During the project, your team should know how to communicate with you, the client, and each other. If you leave management of communication channels to chance, you wind up with an unpredictable and unstable network in which some people collaborate wonderfully and others not at all. This isn't a good situation. You want everyone on the team and outside of the team to know what to expect, how to talk with each other, and what the communication pathways are. Without this kind of clarity, information gets lost, and your project loses time and efficiency while trying to make up for it.

■ **Note** The perception of good communication can be as important as the communication itself.

You face two different client-communication problems. The first is the client not knowing what is going on or how to find out. The second is equally deadly: the client *perceiving* that they don't know what is going on. It's interesting to note that it's usually easier and quicker to bring someone who was confused up to speed than it is to change someone's perception about how in-the-loop they are. And when clients perceive that they aren't being communicated with, they tend to expect the worst. Good team and project communication prevents the client from thinking bad thoughts without knowing the real story—or at least who to ask for the real story. Ensuring that the client doesn't jump to conclusions, and knows how to get the information they need, is at least half the battle.

Finally, good communication has a positive impact on you and your career. Many executives you deal with have only a passing interest in your project, but they have a much deeper interest in you. They want to see how in control of your project you are, how well you can work with upper management as well as your team, and how you handle clients. In short, they're trying to evaluate you for future opportunities, based on how you handle the current one. Because they rarely see you, and almost always in a communications venue, this is the best way to impress them—and the best way to disappoint them, as well. If you're seen as a good collaborator who produces good results, then you may be given a better project, a more impactful part of the business, or more opportunities next time. Life isn't always an audition for something else, but in this case, you may be evaluated on something other than how your project is doing.

Summary

There is no task more important to a project manager than communication. It is the only means by which we can manage our teams, inform our clients, and keep the entire project community informed about what is going on with the project. More than anything else, a project manager communicates; it is therefore important that we do it well.

Communicating well requires a good understanding of two different concepts in communications: audience and channel. You will be working with people who have different affiliations with the project. This includes full and extended team members, other departments, your client or sponsor, and executives. Each group requires a different set of information; it is difficult to craft one message that is applicable to all. Similarly, each group may want their information to come to them in different ways. Some people may prefer to read information in an e-mail, while others may want to be informed in person. Some may want to interact with you while you are delivering content, while others may not. Knowing your audiences and the best channels to deliver your information to them greatly improves the effectiveness of your communication.

The way to best understand your audience and the proper channel to reach them is to treat your communication like you would any other deliverable. You should be gathering requirements, testing results, asking for feedback, and making changes along the way. After you “launch” a message, you should be measuring its effectiveness, and seeing how you can do better next time. This way, your carefully managed communication is no different than something you are building for the project itself. Making sure that your team is aligned is too important to not do intentionally, and this strategy will ensure that you do.

Beyond the value to your project, there is significant value to yourself and your career when you communicate well. Many things happen to projects in the course of their delivery. Some get cancelled, some change focus or shape, others launch with little fanfare or success. But that isn’t the only way that you can make an impression on the people you are working with. A project community that stays well-informed, well-aligned, and delivers on communications that are effective and well-received is a sign of a good project manager, and one that will get more work in the future.

Develop a Plan

“Let’s just get started—we know what we need to do.”

I can imagine an experienced project manager reading the title of this chapter and deciding to skip ahead. I’m sure you think you know how to plan a project, and you probably have the documents and scars to prove it. But to a no-drama project manager, planning isn’t just about creating a schedule, a Gantt chart, or a work-breakdown structure. Well, it’s also that; but planning is mostly a mindset, and it’s possible to have all the artifacts of a plan, such as a schedule, without committing any acts of planning.

■ **Note** Planning isn’t just about creating schedules.

It’s also possible to have the correct frame of mind but not be terribly skilled at putting it in whatever project-planning software your organization uses. Ideally, you would be able to do both, but it’s much more important to have the proper focus than it is to have the proper printout. As discussed in Chapters Three and Four about requirements and priorities, making sure you know what you should be doing is critical to ensure that you’re doing the right thing. Seems obvious, I know.

They key differentiator is to act intentionally. About a year ago, a project manager on my team was running a project that was essentially a complete rewrite of an existing system. Almost everyone from the engineers to the

business owners was sure they knew what to do and that we should get started working. This conversation took place, nearly verbatim:

Project manager: “What should the system do?”

Business owner: “It should do what it does now, but with new features that we’ll figure out as we go.”

PM: “Okay, well, what does it do now?”

BO: “Just make it work the same way.”

PM: “And how is that, again?”

This discussion went on for much too long, with the business owner insisting that the best way to start the project was to rebuild the old system using newer technologies while we determined what to add to it. Then, the owner said, when the new system was ready, it would be easier to add new features on top of the new platform. Making things worse, the engineers agreed. Through some mix of being tired of dealing with the old system and a belief that they knew what the system did, they combined forces and got the project moving.

My project manager was outnumbered and outvoted, and eventually he gave in. Against his instincts, he wrote a requirements document that essentially said, “Make it work the way it currently works,” and a test plan that wasn’t much more than “Make sure nothing breaks.” As program manager, I caved as well; the business owner basically refused to give us requirements, and the engineering team was at the starting gate trying to push their way out. Better to open the gate and hold on tight before the horses leave without us—or so I thought.

I should have listened to the project manager. The engineers got started building while the business owner and the project manager began working on the new requirements for the system. Because we weren’t rigorous about figuring out what the new system was supposed to do (beyond “not break”), it was very difficult to manage its construction. And as it turned out, the engineers didn’t know how the system worked.

When it became clear that the rebuilt system was breaking things and was slower and throwing more errors than the old one it was intended to replace, the project manager had to disengage from specifying new requirements to focus on repairing the construction. As you can imagine, this was just short of disastrous. He had to go back and find all the use cases and details for the system, begin testing them, and figure out what needed doing. By the time the budget for the project ran out, he had wrestled it to the

ground. The newly built system was as functional as the old one, and it was in much better health as a system.

But we still don't know everything it does, and now we may never know. I hope this is the last time I make the mistake of allowing a project to proceed without clear intention, but I'm doubtful.

Acting Intentionally

I often state that decisions made on purpose are usually the correct ones. This is true for everyone and every decision. As a project manager, it's more important because you're making decisions not only for yourself but also for your team and sometimes for an entire organization. The worst thing you can do is let things happen by accident, or let entropy and inertia take control and decide for you. Some efforts are small enough that you don't need to actively manage them. But in nearly every project worth doing, at least one decision should be made intentionally, even if it's only to decide to work on the project in the first place.

In order to behave this way, you need a firm grasp of what you're doing and why you're doing it. In the previous example, no one on the team was certain what they were doing or why. They only knew what they were working on. They had no good idea what they were trying to achieve or what the correct result should look like. Worse, they didn't know when they were done. This was the result of acting *coincidentally*, not intentionally. That is, the work they were doing coincided with a handful of outcomes, but it was never clear what the ultimate result should be.

This is a major source of drama in projects. It's possible to have a team of people all working diligently on their deliverables without worrying about the bigger picture. Eventually, they want whatever they're working on to fit into the puzzle; they want to visualize progress and understand their place in the world (or the project). The client wants to see these kind of results as well. A client once said to me, "I see a lot of bricks, but I don't see a building yet." He was right. We had built a lot of bricks—pieces of functionality, large and small—but we had yet to put them together in any meaningful way. When we did start assembling them, we discovered that we could have stopped much earlier—we were finished with the project and didn't know it.

■ **Note** Without knowing what you're doing, you won't know when you're done.

Making decisions intentionally seems like an obvious statement. But all too often, major decisions either aren't made actively or are left until only one possibility remains. This kind of passive decision-making is rarely done out of laziness, but more out of uncertainty or insecurity. Few people are energized by making decisions before they're needed.¹ The problem is that it can be hard to recognize when decision time is at hand. When the time to decide has passed, you're left in a confused state or with a limited number of options; or maybe you're in a corner where no options remain, and you have to work your way out.

Removing drama from projects is sometimes about removing confusion and uncertainty. Being thoughtful in your direction-setting helps eliminate confusion and leads to a much more peaceful, and usually productive, project team.

Scheduling and Traditional Planning

The last thing I want to do is impugn traditional project planning and scheduling. I have seen many projects saved by a solid plan that the team followed and updated religiously. A well-executed work-breakdown and dependency plan can be a thing of beauty. And if everything goes the way it's supposed to—that is, it goes *according to plan*—then eventually the project manager no longer needs to actively manage anything, and the schedule drives the project to completion.

However, traditional planning suffers from three weaknesses that this chapter discusses in the context of running projects drama-free: diminishing returns that sometimes happen when you have a zealous project manager, false information that plans sometimes provide, and inflexibility when plans are taken too literally. You can't have a project without plans and schedules, but it's possible to create them in such a way that they detract as much or more than they help.

Overplanning Project Managers

The basic artifacts of a project plan must be in place. This means things like Gantt charts, a work breakdown structure (WBS), a dependency chart, vacation schedules, and shared lab space time. Additionally, as an academic exercise, these elements the only way to answer fundamental questions like, “Do I have enough budget and resources to complete the project?” and

¹ And you should probably watch out for people who are.

“When will we be finished?” Such artifacts are usually detailed down to the hour, scenarios are planned, and risk sections are identified. All important tasks, to say the least. But this is no way to run a project.

■ **Note** Managing a project isn’t the same as managing a project schedule.

The level of detail required to answer these fundamental questions is the wrong level of detail when you’re trying to manage the day-to-day operation of a project. And therein lies the dilemma. Proper management of a project requires tasks, milestones, and deadlines. It shouldn’t be so overly prescriptive of people’s time that taking a 90-minute lunch shows up on the project plan. And yet, frequently, plans are measured in minutes, with tasks listed in 30-minute or smaller increments.

This doesn’t have to be a big deal, as long as the project manager isn’t overly restrictive about the minutiae of the project schedule. When someone asks questions like, “Are you 87% complete or 89% complete?” or constantly updates the project plan when tiny slippages occur, then you wind up with the exact wrong outcome. People begin avoiding giving real status updates, because they don’t want to deal with the anxiety associated with the fallout, and eventually everyone is unclear where the project stands. And as you know, lack of clarity is a primary driver of drama.

False Information in Plans

Although lack of clarity caused by a reaction to a micromanaged project can create drama, it may be worse when the project manager thinks their academic exercise is real and a good indication of how the project is doing. This type of project manager asserts that because all the project tasks are 60% complete, as per the plan, then the project is also 60% complete; or that the project is on track because nothing has yet happened to cause the ship date to change in the project plan. If I ask a project manager how a project is trending toward its intended release date, I expect them to know what things are ahead and what are behind, and to have a sense for how the project is doing. If they have to consult their project chart first, I know we’re in trouble.

A long, long time ago, I was a team member on a project whose manager ran it exactly this way. Her project plan was a thing of beauty, and it was printed on a 2-foot × 3-foot piece of paper and displayed on the wall of the war

room. Whenever changes were made, she printed a new plan and took down the old one. I was inexperienced enough to think this was a good thing.

The project's release date was in the distant future, and few of us on the project could understand it. We all felt that we were making good progress and might be able to go beta within a few months, not a few quarters. But the project manager was very firm that the ship date was at least nine months out, and she had a detailed plan that showed why. It was difficult to dispute.

Then one day, an engineering director came into the war room and looked at the plan. He too thought the project was going well and wanted to understand why the ship date was so far away. Looking at the plan, he noticed something incredibly trivial on the critical path that by itself was causing a four-month delay in the schedule. The task was something like, "Get key-card for server room." The director picked up the phone and talked for a few minutes to someone in IT. He hung up, flipped his card on the table, and said, "Here, take mine. They'll issue me a new one."

The next day, a new plan was up on the wall: it showed a release date six months sooner. I never saw the project manager again, and the plan on the wall was never updated again, either. However, for the rest of the project, our perceptions of how we were doing was much more aligned with what our project manager was telling everyone else.

■ **Note** If your team is more optimistic than your schedule seems to be, the problem probably isn't with your team.

Inflexible Plans

The final weakness that I often see in a project manager who over-plans is a high degree of inflexibility when it comes to the plan. As program manager, I'm often forced to move people from one project to another. Sometimes it's a temporary reassignment; other times it's a full shuffle of project teams. I have seen project managers insist that removing one of their engineers (even in a team of ten or more) will cause a day-for-day slip in the project. This isn't the way to win my trust.

But this kind of statement is provable. For example, a project manager removed a person from the project for a day, and the software they were using indicated that doing so would indeed push out the final release date from Monday to Tuesday. This kind of inflexibility wasn't helpful. I then asked,

“What happens if I remove all ten of your engineers for a day?” The same thing happened. If I took one or ten engineers off the project, the ship date moved out by one day. The project manager wasn’t pleased. The software told him exactly what he was asking, but his brain was telling him otherwise.

In the end, he figured out how to release the engineer for a few days to do something else I needed. But rather than considering the request² for a few minutes and deciding he could accommodate it, his reliance on an inflexible, brittle plan nearly made us reach the wrong outcome.

Of course, you must make plans. That’s what you do. And of course, having a plan for how to execute is significantly better than the opposite. As long as you recognize that managing a project isn’t the same as managing a plan, and you understand where you are in the project based on higher-level concepts, you’ll do fine.

Making Other Plans

Project schedules aren’t the only kinds of plans you need to construct. You need to make decisions ahead of time about many things regarding situations that may occur as the project progresses. You may need to haul these out and put them in motion if something does happen. Let’s look a few examples and talk about why they’re important to consider.

Earlier, we talked about making intentional decisions and that when you do so, you’re much more likely to make the correct choice. In certain instances, you want to make choices not only thoughtfully and on purpose, but also before the decision point occurs, allowing you some distance to make a call without the pressure of the situation upon you. It can be irritating to take a few days to decide how to put out a fire, even if that gets you an optimal answer.

Three types of plans are common to many projects: staffing plans, communication plans, and risk-management plans. Not every project has all of these, and many projects have lots more than these, but this sample highlights some of the major challenges you face as a project or portfolio manager.

Staffing Plans

Ideally, every project has a staffing plan. Using no-drama terminology, it means thinking ahead of time about the skills the project requires to be successful,

² It wasn’t really a request, but it may have seemed like that to him.

how long they're needed, and what training is required or what outside skills must be brought in. It shouldn't include people's names, nor should you take advantage of knowing that one of your database guys is also a whiz at website user interfaces. This plan represents your *full request*.

If you're able, match this up to the client's budget, if there is one. This can be measured in dollars, hours, or some other unit. If the client or your organization has no concept of budgeting, try to make sure the value you provide matches the amount of resources you're requesting. If I were your program manager, I would ask you to do an analysis comparing the level of staffing you're requesting with the value you're providing—please have a good answer.

Finally, you should think about scenarios. What happens if the program manager give you only half of your full request? What if they give you all of it except a single scarce resource? What if they give you more than you're requesting? Or, the more likely scenario, what if they ask you to take on someone junior, because that person needs to learn something your project can provide? You should be ready to answer these kinds of questions if they arise.

Why is this kind of staff planning important? As much as I'd like to say that people are people and everyone is the same, they aren't. Every project manager has one or two favorite team members, artists, analysts, engineers, or the like, with whom they feel a certain rapport and can get more work done. You must assume that you aren't going to get them on your team, and plan accordingly. Additionally, you need to be prepared for the types of resourcing games that portfolio managers are forced to play. Because they have a target ROI for their entire portfolio, they may over-invest or under-invest in your project.

Here's how to avoid taking it personally: think ahead of time about what your contingency plan will be if your ideal plan doesn't materialize. If you've already thought of this possibility and have an idea what you'd do, then getting a less than ideal staff is much less likely to cause stress on you and your project. Conversely, if you don't do this exercise, then your team will know you aren't happy with the outcome, and whatever anxiety you feel will greatly affect the project.

Risk-Management Plans

Risk-management plans are stressful by their nature; they assume something will go badly. Your task is to think of everything that can go wrong with the project, determine the degree of likelihood and impact, and then come up

with a plan of action if those risks do materialize. Even better, take steps now that will help you avoid risks before they happen. In other words, solve problems you don't yet have.

Figure 10-1 shows a simple risk matrix. It includes two values, high and low, but I have seen project managers use from three values (high, medium, low) to a 100-point scale. The matrix plots the likelihood of a risk occurring versus the impact it would have on the project. Whichever scale you decide to use, it's important to think about your risks using such a matrix. When something comes up and you need to communicate it, you should be able to describe where on this chart it lies.

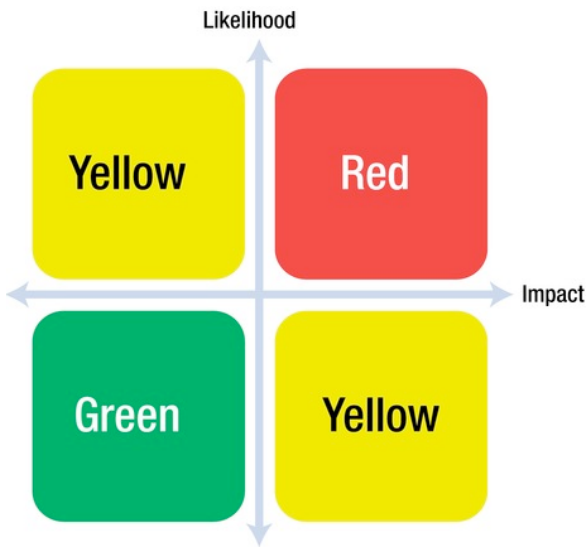


Figure 10-1. Risk-management matrix

This kind of thinking removes drama in two ways. The first is the simple communication tool provided by a 2×2 matrix. You, your team, and your client can all agree where things fall on the matrix and easily determine whether you need to worry. The more complicated your scoring system is, the less obvious it will be to someone who isn't as knowledgeable about the details. And clarity of communication is key.

Secondarily, having the plan admits two things: that you expect something to go wrong and that you're ready for it. Both of these statements relieve your team and sponsors. So many things go wrong in daily life that you almost expect something to come along and ruin everything. You need to prevent that

kind of thinking from happening. You must show your team that you're prepared if something goes wrong—and it will—and don't intend to let it make your project fail. This kind of attitude goes a long way in solving problems.

■ **Note** The right attitude sometimes can limit drama better than avoiding it entirely.

Communication Plan

The last kind of plan is very important in all phases of the project: the communication plan. It seems silly to new project managers that they need to communicate a plan that expresses the plan for how they will communicate. And yet, you most certainly do. Deciding who will receive communications, what form they will take, and how often they will happen is vital to maintaining proper visibility into your project. You don't want someone to think they're in the dark about your project, because when that happens, they invariably imagine the worst.

How do you make a communication plan? You start by talking to those with whom you plan to communicate. Chapter Nine discussed asking stakeholders what kind of communication they want, who should get communications, and all sorts of ideas for how to get in touch. You learned about following up to make sure your communications are what stakeholders want and need in order to stay in touch with the project at the level they desire. You use this plan to confirm such things with them.

When someone says they aren't getting enough information, or the correct kind, or on time, don't get defensive or make excuses. Doing so drives drama. Instead, point at your communication plan and explain that you were following it, but that it clearly isn't accurate. Promise to update the plan based on this new understanding, and follow the new plan going forward. That doesn't mean you should say, "I was only doing what the plan says to do!" Although that may be true, it isn't helpful.

But without this plan to support you, you have no choice but to apologize or make things up. And without a plan, you may forget to change the way you're communicating, or your communications may be erratic or difficult to understand. It's wonderful to give clear and concise updates to everyone who wants them, but it's more important to have a plan, stick to it, and adapt when it's not working.

Sticking with the Plan

Having a good set of plans is a great way to avoid drama in your project. Whenever something seemingly unexpected (and bad) comes up, there is tremendous power in being able to say, “We thought of that, and we have a plan for it.” This kind of statement takes all the wind out of a problem’s sails and turns people from focusing on the problem to curiosity about what the resolution will be. The act of not being caught unaware brings a measure of comfort to your team and your clients, even if your plan doesn’t turn out to be that great.

However, if you create a plan to manage risk, communications, or anything else, and you don’t consult it, you may be worse off than if you didn’t have one at all. This means you thought of a possibility before it occurred, and you decided what you would do if the situation arose, but when the situation did arise, you didn’t use your plan. Rather than appearing unprepared, you seem foolish, which is much worse than simply being wrong. You can legitimately tell your program manager that months ago when your plans were drafted, you were unable to think of everything. That’s okay. But panicking your team, your client, and your project is a much harder situation from which to recover.

This doesn’t mean you’re bound to a plan you created months ago using incomplete information. You bring out the plan and make an active decision about whether you should still follow it. Months ago, you probably thought of this contingency, determined a course of action, wrote it down, and had it approved. In the time that has passed, though, things have changed, the environment may be different, and you’ve learned a lot. It’s not awful for your plan to no longer be the right course of action; but if that’s the case, you must consciously reject it.

■ **Note** Even the decision to change your mind should be done intentionally.

This goes back to intentional decision-making. If you decide not to follow your preexisting plan, you need to do so assertively. Think about the following statement: “We had a feeling this might happen, and we planned what we would do if it did, but now we don’t think that’s the best course of action. Now we think the best thing to do is ...” This shows you weren’t caught unprepared and that you’re in control of the situation. Your confidence in your actions is what everyone needs to see.

So, something comes up: a risk, a change in staffing, or any of a lot of other things. If you don't have a plan already in place, it's no big deal—you can't think of everything. But if you did think of it, all the better, and even better if your plan turns out to be the preferred solution. However, don't be so inflexible that you proceed with a plan that may be outdated. Consider whether it's the right course of action, and if it isn't, actively change your mind. Allowing problems to run wild, or deviating randomly from preexisting plans, may make your team, your client, and your program manager lose faith in your project.

Summary

Planning is a primary competency of project managers—sometimes so much so that the amount of time you spend making plans can frustrate your team or your clients because they perceive a lack of forward progress in the project. It's your job to explain the value of knowing where you're going before you start running full speed. If you don't map the road at least a little before you accelerate, you won't have much control over where you wind up.

Planning your route is one strand of a larger concept: making intentional decisions. It's very easy to just let decisions happen, or to delay making them until all your options are gone. It's also easy to never make decisions in the first place and let a team member choose whatever turns out to be easiest for them. When you make decisions on purpose, you almost always make the right one. When you don't, you may still wind up in the right spot, but only by accident, by luck, or through more work than you should have spent making the call.

You have to make many types of plans for your projects. However, it's important to recognize that you're managing projects: not plans, not project schedules, and not a bunch of documents. You're managing a client with a goal and a team that needs to deliver. Failing to manage the team or the client, no matter how well you're managing your schedules and project artifacts, leads to a bad relationship with your program manager in the end.

Finally, sticking with your plans is a tricky concept. It's great if you have plans that cover contingencies and that you thought about them ahead of time. The next chapter talks about being prepared and how certain kinds of unexpected problems should be thought about and planned for before they occur. But just because you're prepared doesn't mean you need to use all your preparations. It's much better to have options ready so you can intentionally choose which one to use. Without options, and without making an intentional choice, all your pretty plans go to waste.

Prepare for Problems

“Honestly, we hoped this wouldn’t be a problem.”

When a troop of Boy Scouts decides to go into the woods, they know to bring along things like bug spray, matches, and rain gear. This seems overly obsessive; they don’t know ahead of time that there will be insects where they’re going. And it’s highly unlikely that they can use both their rain gear and their fire-starting equipment; they probably need one or the other. However, this kind of detailed preparation is what the Boy Scout pledge is predicated on. There is no way for them to know what problems they will encounter, but they do know what troubles are likely to happen. They also know that if the bugs start biting, no one will give them any sympathy for forgetting their repellent.

I polled half my project-management staff and discovered that none of them were Boy Scouts. But they know I expect them to act as if they were. In project management, an awful lot of things can go wrong, and many of them are foreseeable, or at least shouldn’t come as a surprise.

Expecting the Expected

The basis of this book is that many causes of project failures are the same over and over again. The original working title was *Expect the Expected*, which

means that many of the problems you face are ones that you could have predicted you would face. Projects can fail for all sorts of reasons. Some of those reasons are interesting and a complete surprise, but they aren't the norm. When things go south for unusual reasons, it adds to the list of stories you have to tell at the next post-mortem. Novel reasons are good.

■ **Note** A lot of project management is simply avoiding the problems that you expect.

Your portfolio manager has seen a lot of these problems before and is likely to be irritated if you try to explain that your project is the very first one so affected. I would venture to say that at least 80% of the “unexpected” problems a project faces were expected—you hoped they wouldn't occur. As I've heard said many times, “Hope isn't a plan.” If a project manager falls prey to the problems nearly everyone faces, it's not hard to say who is at fault.

You, as project manager, may not be blamed by the client, the sponsor, or the team. The portfolio manager will work hard to get you off the hook and make it clear that a unique problem sank the project, not the way you managed it. But you and the portfolio manager know better. You could have done better, thought about these kinds of issues before they occurred, and perhaps prevented them in the first place.

Earlier, this book talked about how a well-run project is nearly indistinguishable from a project that is easy to run. But few projects are easy, so if a project appears simple, you probably headed off some issues before they manifested. Although you may not get all the blame if your project fails, you won't get any credit either. And if your portfolio manager can't trust you to you deliver, then your projects will wind up being uninteresting and routine.

I used to keep a list of expected problems, but I eventually gave up. Each passing project added another issue to the list—sometimes more than one. Before projects began, I gave the list to the project team and implored them not to come to me with a project that was failing for one of those reasons. If you're currently a project manager, your portfolio manager will appreciate you failing for an interesting reason, but not one of these.

There are two parts to expecting the expected. The first is knowing what you face in every project and learning not to be shocked when those issues appear. The second is learning how to identify such problems and what to do about them when they crop up, because they will. If you came to me and said, “The problem is...” followed by one of the issues discussed in the

following sections, my immediate response would be to ask what you're doing about it and why you didn't see it coming.

■ **Note** Many of the problems all project managers face are actually the same short list of issues.

Although the list that follows isn't intended to be exhaustive, it gives you an idea about the kinds of problems your portfolio manager won't consider unexpected. Many projects are faced with more than one of these at a time. A few of them are the responsibility of the client or the situation, but a few are things the project manager or the team must figure out how to mitigate. And some sneak up on you and don't seem like problems until it's too late to do anything about them.

Unclear Project Ownership

This is one of my favorite things that a project manager should have recognized before the project began. Chapter 8 talked about stakeholder identification, and how vital it is to a successful project. The primary person the project manager needs to identify is the project owner. If the project manager can't determine who this person is, they should continue trying to identify the project owner and not run the project yet.

Certain organizations have a "one project, one owner" rule, meaning exactly one person must be identified as the owner of the project. It's fine if the project impacts several groups and you need to gather requirements and restrictions from all of them. But when it comes down to it, you want to know who, by name, you can go to in order to get project decisions made. It's pretty hard to overestimate the importance of this.

In the course of the project, decisions need to be made, compromises analyzed, trade-off decisions adjudicated, and general direction determined. As soon as the number of people making these decisions becomes more than one, the time required to decide increases geometrically. One person is preferable, two is manageable, but three turns out not to be three times as hard, but nine times as hard. And when you get over four, you need another project manager simply to manage that group of "owners."

But how does a project manager recognize this? It mostly comes down to decision making. It's normal for a business or project owner not to have final say on the entirety of a project. For instance, if you're running a business-

development project, you don't want the business owner to have final say on compliance issues. However, if the owner is having difficulty making decisions within their own domain, that's a sign that the "owner" isn't really an owner.

Another sign is too many of the business owner's decisions being reversed. They can be reversed by executives, or the owner's manager, or someone from another group. This happens occasionally, and it's not alarming on its own. However, if this happens more than once, team members start doubting that "final" decisions are actually final, and they don't treat them as such. Or they ask for validation of decisions before they begin working, which is just as damaging.

One way to handle this is the same approach salespeople use when they're trying to sell a product. You ask, point blank, "Are you able to make this decision yourself, or do you need to check with someone first?" Then you need to listen for the tone of the reply. If it's a weak "Yes," or something noncommittal, then you may have a problem that you need to follow up on. If you get such a firm "Yes" that it covers areas you don't believe—"Of course I can decide on tax law issues!"—then you should also beware.

If you find yourself in this situation, it's an excellent opportunity to whisper it to your program manager. This isn't escalation; it's asking for help. If you think your business owner—your prime decision maker—is unable, unwilling, or unempowered to make the decisions you need them to make, then that's a problem you must address. Usually, this can't be done at the project-manager level; your program manager and the business owner's command organization must deal with it. But before you act on too many shaky decisions, be sure the decision is being made correctly, and by the correct person.

Slow Turnaround

Turnaround may not be the right way to think about this issue, because slow turnaround is an effect, not a cause. By *turnaround*, I mean the length of time it takes to complete anything that needs to be done by the client or business owner: deciding which logo to use, determining budgets or expected launch dates, and so on. While you're waiting, your team drifts in a random direction, or you set a direction by making a best guess, or the team does nothing. You're gated by the client and acting inefficiently while you wait.

This problem is similar to unclear ownership but manifests itself differently. In general, it means one of two things: prioritization issues or uncertainty about the project's future.

Assuming you have a good project owner who is empowered to make the necessary decisions and perform the necessary actions, then you're most likely looking at a prioritization problem. The project owner may be doing several other things that they believe are higher priority. This isn't necessarily an incorrect assessment on their part; everyone has multiple things on their plate at any one time and must decide for themselves which are most important. Sometimes that means you get slow or nonexistent responses, or responses that don't appear to be well-reasoned or thought through. They may know they need to respond to you, but they don't have time to do it well, so they do the best that they can with the time available. Depending on how little that time turns out to be, you may get an answer that is barely any use. This is a big warning sign.

But it's much better to find out your project owner or client thinks the project is important, just not as important as other things going on, than it is to learn that your project has an uncertain future. Sometimes, the cause of slow response time is that the project owner is thinking of halting the project. Rather than giving you an answer and potentially more work to do, they're sitting on your request. If you're a consultant, the owner may not want to incur any more billable hours; otherwise, they may feel guilty about you wasting effort on a project that's headed for cancellation.

■ **Note** Slow response times mean that either the client has more important things to do or they think your project isn't essential.

Sometimes this issue is hard to spot. You regularly find yourself waiting for the client, but not often enough to cause alarm. Or they give you a plausible excuse, such as waiting to talk to the CFO, who is on vacation for two weeks. Or your program manager asks you how things are going, and you don't have an update since last week or month, because you've done nothing but wait for something to be delivered to you.

Again, this is the time to tell your program manager something like, "I can't get the client's attention." Because that's really what you need—you need the client to pay attention long enough to give you whatever is currently stopping your progress. Or you need them to tell you to stop working. An honest answer that the client is busy with something else and won't get back to you for a few months is fine; you know how to deal with project delays and rescheduling. What you can't deal with is thinking for weeks or months that something will arrive "day after tomorrow."

This kind of miscommunication can cause an awful lot of drama in your team. They think they're waiting for one piece of information or decision, or permission from the client, and then all will be fine. In the meantime, progress is stalled, milestones are missed, and the project schedule looks more and more scary. Either you or your program manager must recognize when this is happening and have a heart-to-heart with the project owner. The inclination is to advocate for your project, but that's the wrong approach. Instead, advocate for the client and let them know you understand: priorities change, people get busy, and emergencies happen. You need to know whether you should continue working, delay for a few months, or stop.

Approaching a client like this often yields very interesting results. If you don't feel comfortable doing it, then you have your program manager do it. Either way, it must be done before you drive your team crazy with inaction.

Lack of Fundamental High-Level Agreement

Back in Chapter 10, we discussed the peril of moving forward without actually knowing where you want to go. Sometimes this is okay. You don't need to decide if your new product will be red, blue, or purple until later. But it's possible to attempt to move forward before gaining alignment that the new product should be produced at all. Doing so is almost always a mistake.

I once worked for a company that was very successful at making low-end, low-cost products. We sold a lot of these items at prices that couldn't be found anywhere else. But there existed a natural upgrade to one of our products that would cost several orders of magnitude more than what we currently sold. After some discussion, we got the go-ahead to build it out.

Even though we had approval, we didn't really have alignment with the executive team. Some of them thought our service model would have to change: if we were going to sell a high-end product to our low-end customers, they would demand a different level of service. Some of the executives were willing to give the product a try, but because our customers were low-end buyers, they didn't believe those customers would upgrade to the offering. A few others were excited because we could offer this higher-end product at a lower-end price and it might dominate the market. We had no high-level alignment on why we should create this product or what it would mean to the company.

Along the way, we ran into some meaningful challenges. We needed to change virtually everything about the way we did business, but only for the customers who bought this product. We built out instructions and self-help,

tutorials, and human help in order to educate customers and interest them in buying the product. Eventually we gained some traction, but not as much as we hoped. Perhaps the executives who said we were targeting the wrong market were correct.

Whatever the cause, we launched a product that was two or three levels above our other products. And we did so without knowing what we would do with it, how we would support and market it, and how we would maintain it. As a result, we did none of those things, and sales were disappointing. The program manager in me looks back and says, “Of course sales were disappointing. How could they be otherwise?”

You can anticipate this issue at the start of the project. When you’re interviewing your client, sponsors, and stakeholders, try to get at the key question: “Why are we doing this in the first place?” If you get as many different answers as people you ask, then you don’t have anything approaching alignment in your project.

■ **Note** If a project’s sponsors all have different reasons for wanting it, then you may still be okay, but you don’t have alignment.

Sometimes this is okay. Just because multiple executives have different reasons for wanting a project doesn’t mean you should do it. That is, as long as everything goes well. As soon as you need to make tradeoffs and prioritization decisions, you begin eating into an executive’s personal success criteria, and you may wind up losing their support.

As project manager, you can try to gain the high-level alignment your project doesn’t have. This falls somewhere between being a tall task and an impossible one. If you think you have a chance of getting all the possible sponsors on board, then your program manager will allow you to try—but won’t expect you to succeed. More important is understanding the misalignment and being open and honest about how your decisions may cause you to lose the support of certain people.

In the case I described, we decided not to invest in human beings to help support our high-end product. Instead, we relied on videos and other self-service type aids. These were better than nothing, but no one thought they were good. They did have a very good ROI: we could build them once and never pay for them again, and they provided a basic level of help to our customers.

When we made this decision, we knew we were running the risk of losing our head of client services as a supporter of the project. And we did lose her, practically before the project launched. We made a calculated decision that if the product wasn't a success, the self-help tools would be sufficient. But if it was a success, then we could afford to invest in more staff and more support for our customers.

It wasn't a success, and many believed it was a self-fulfilling failure. Our decision to wait to see if it was worth it to provide customer support for the product wound up ensuring that the product wasn't well received, which in turn caused us not to invest in any more support for it. Seems pretty obvious when you look at it that way, but at the time no one did. Least of all the project manager.

Skill Gaps

This problem is a lot more tactical. It's not uncommon for a project team to be missing one or more critical skills that are necessary for a good outcome. Often you staff a project without knowing exactly what you need and what the team has to know in order to deliver. Sometimes you make the explicit decision that although no one on your team possesses the knowledge you need, you can put smart people on the team and they'll figure it out.

As project manager, you should know the cause of a skill gap. Did you not know what experience was necessary, and you guessed incorrectly? Did you decide that training your staff would be part of your project? Is the resource you need so scarce that you decided your project wasn't worth it and you could make do with whatever you got? Or did you hope your team would figure it out? As a program manager, I've gone each of these ways several times.

Depending on the reason, you need to respond differently, which is why you must understand the project at a deep level. If it's part of your responsibility to train your staff, then you need to build that into the plan. If you're expected to make do with what you have, then you have to factor that into your plan, as well. If you think your portfolio manager made a mistake, well, maybe they did. But you should ask about that, too.

■ **Note** There are many reasons your team may be lacking the skills you need. Some of happen on purpose.

So you find yourself managing a project without the critical resource, personnel or otherwise, you need in order to be successful. You have a couple of options. You can ask for another resource or a personnel swap. There is a chance this could work; but more likely, if your program manager wanted you to have someone with that skill, you'd already have them. You can adjust your schedule in a way that's transparent to both your manager and your client, to compensate for the lack of the skill. There is risk in this, however, because you "don't know what you don't know." You have to guess how long your team will take to learn the necessary skills. Finally, you can refuse to proceed without getting an addition to your team. Play that card wisely.

No matter what tactic you decide to take, you need to talk with whomever approves project staffing as soon as possible. Find out whether the person is missing due to an oversight, because of a lack of knowledge about what the project needs, or on purpose for any of a variety of reasons. Ideally, this person is also interested in your project's success, so you can plot a course through the problem. If not, take the most conservative stance and assume your project must proceed with unskilled team members. Scope and schedule accordingly. Your job is not to mislead or lie to the client and tell them all is well. Your job is to be honest and let them know that although you're optimistic, you think this is a risk for delivery. This message may get around, which is okay. A little honesty can often solve a lot of problems.

Breaking the Process

As many engineers are happy to explain, processes exist for a reason. They cover years of experience, and they attempt to mitigate unseen and unknowable project risks by their very existence. This accumulated operating procedure is the result of many projects in the past that ran into some problem or another, after which a process was added to avoid that specific problem in the future. Some processes are old enough that no one remembers why you do things the way you do. All you know is that when you follow the process, things go correctly.

But the process has a weakness: it's slow, it's cumbersome, and it usually feels like overkill. Little or none of it seems to apply to your project or situation. And often, when asked, the process engineers can only provide outdated or unspecific reasons why you should follow the process in the first place. Because everyone follows it, you don't have examples of what happens when you don't.

As a project manager, you're always looking for ways to save time and effort by eliminating low-value tasks or tasks that don't apply to you. Some of

these procedures appear to fit the bill: they aren't useful in your case, and they don't provide much protection, so you may as well skip them. In other words, you're probably about to make a mistake.

Years ago, when I worked for a data-feed company, we had a very lengthy, very ridiculous process for acquiring a new feed and getting the content into our main feed. No matter how much content we were talking about, the process was the same. I was running a project to integrate a very small, very niche data source from Canada into our primary product. The process to get this feed into our core product was longer than the project was supposed to last. Instead of having a healthy respect for the process, I laughed openly at it; we were expected to have the new feed in place in 12 weeks, and the process took 15. Obviously, I thought, the problem is the process, not the project.

Not only did I decide to ignore the process, but I was certain I was right. So certain that I was able to convince most of our engineering staff that my project was special and should have different rules. The process wasn't meant to be used in situations like mine, and I should be exempted from having to use all of it. Maybe we could create a new process specifically for future such projects so they wouldn't be saddled with an archaic checklist to get simple work done.

Or maybe I could have my project ignore the right way to do things and totally mess things up not only for itself but also for anything that touched the new product. The full-blown way to do things was full of checks and balances—safety mechanisms I decided weren't needed for my project and that we'd be fine without.

I suspect you can see where this story ends.

■ **Note** Your project isn't a special case.

The feed didn't work, and the system was less stable now than before I began, because we cut a bunch of corners. This was the first time I heard the fateful question that should be the mantra of a no-drama project manager. My boss asked me, "What did you think was going to happen?"

Ignoring the approved way to do things was a risk, but it was a risk I was willing to take because I misunderstood it. I was encouraged by my ability to sell the idea to the staff, and I convinced myself that even though we had a

procedure that exactly applied to what I was trying to do, I could safely ignore it. I couldn't have been more wrong.

I also had no good answer to the question from my boss. I was about to reply that I thought my project was different and would go fine. Part of me was ready to say that the project was going better *because* I had ignored the lengthy and expensive process for getting things done, which moved my project into having a much higher return.

But then I envisioned his follow-up question: "What evidence did you have that suggested to you that you were right?" The answer was "None"—I had no evidence whatsoever that skipping the operating procedures was a valid thing to do. All I knew was that I was saving project hours by not doing things in the tried-and-true fashion.

Additionally, this caused anxiety in my team. Although they were ready to admit that I might have been right, and we could safely do it the way I said, they weren't sure. But they were sure that if we did it the correct way, the project would have a good result. I was working with seasoned professionals who knew their jobs; I was the one who didn't know what he was doing.

In your projects, you may be tempted to cut corners, particularly ones that you don't see as adding value. But some of those corners exist for a reason, and often that reason isn't readily apparent. The excuse "I didn't realize" or "I didn't know" won't work with your client or your program manager, and probably not even with your team. If you didn't know, then why did you act on that lack of knowledge rather than finding out or keeping things the way they were? Short of having a reason backed by evidence or a strong hypothesis, you're likely taking a risk that will cause you problems in the end.

Underestimating Complexity

Most projects are trying to do something complex. If they weren't, they wouldn't need project managers. You aren't hired to manage routine tasks. It's the nonroutine tasks that you're hired to deliver. It's true that some of the greatest projects happened because people either ignored or chose to overcome an immense amount of complexity, difficulty, and complication. But most of them didn't have a client who was paying by the hour, or a business that was eagerly awaiting—or had its future riding on—a new product launch.

One of my least favorite phrases to hear from a client is, "Can't we just...?" This is usually followed by an inane oversimplification, such as, "Can't we just put a note right above that button and tell people not to push it?" Or,

“Instead of buying a new size of shipping carton, can’t we just use one of the bigger ones and stuff it with newspaper?” These examples and other, equally jaw-dropping ones have actually been asked of me. This is what makes me sad when I find myself or my project managers doing the same with teams.

■ **Note** The answer to “Can’t we just...?” should default to “No,” unless proven otherwise.

To be fair, one of the roles a project manager has to play is to both limit and manage complexity. One strategy for doing so is to suggest simple solutions and see if any of them stick. I recall the movie *Dave*, in which Kevin Kline pretends to be President of the United States. Kline is trying to find a way to generate extra funding for a children’s program, so he decides to pour a bunch of money from the Treasury into a regular savings account in order to earn interest. The Secretary of the Treasury is first very confused but eventually admits that this idea may work. Kline generates more than \$100 million doing something I taught to my son when he was eight years old.

What the movie doesn’t show is that although this approach may be technically possible, it’s difficult to actually do. In order to keep the money FDIC insured, they would have to find hundreds or thousands of banks to put the money in. Because of that, they would probably need to hire several people to deal with the statements and details of opening these accounts and making sure they weren’t lost. The ongoing cost of this “simple” solution would likely be much more than originally anticipated.

But when you’re trying to be creative with solutions, often you’re focused on the solution, not the implications. Your simple solution may indeed work; but worse, it may appear to work at the beginning but crumble at the end. One sign of this is when your team, client, or manager is more skeptical about your solution than you are. Another is when you start hearing replies like those Dave heard: “I suppose that may be technically true....” It may mean you haven’t appreciated the actual complexity of your problem space and are one mishap away from failure.

The drama this can create within your team or with your client isn’t worth it. If your team begins to doubt that the solution they’re building will work, then they may have misgivings about delivering it. Your client may be more worried about the long-term impact of what you’re doing than about the speed with which you reach a solution. And the people who have to implement your idea (those suckers who had to open all those bank accounts) may be silently cursing you and your project.

■ **Note** People are generally skeptical of simple solutions to difficult problems.

It's okay to be optimistic about your decisions and projects. But it's not okay to be blind to the challenges that reside within the problem you're trying to solve. If you do go down this road, someone, somewhere will ask you, "Well, what did you think was going to happen?"

Risk Introduced by Schedule "Make Up"

Your project has fallen behind schedule. Not uncommon, and not all that unexpected. Your client or program manager asks you for strategies to "make up" the time and get the project back on schedule. And, of course, the client doesn't want to trim any functionality. They want everything included that they were originally supposed to get. In other words, to meet the schedule, they want you to work faster.

But you realize that asking everyone to work late isn't good for the team; and worse, unless you get them to work very late for a very long time, it won't make up the shortfall. So, you start to look for things to cut—cuts that introduce risk to the project but may not be noticeable by the client. Perhaps you decide that your project doesn't need to be load tested. You rely on the skill of the engineers to make sure performance is good. Maybe instead of sending every single permutation and combination through quality assurance (QA), you only send a representative sample. Or perhaps you take on some of the tasks yourself, such as copywriting or graphic design.

Whatever way you choose to make up time in your schedule, you introduce risk to the project. You're either reducing quality control or having tasks done by people who aren't skilled to do them. In essence, you're hoping that those steps weren't needed in the first place and nothing bad will happen.

The alluring part is that some evidence supports you. Perhaps the last several projects you ran did fine under load. Maybe you think you did a good job picking representative examples of what you want tested, and you're convinced you've covered all the major possibilities. Or perhaps you've substituted one deliverable for another of significantly less cost, but also significantly less value. In the ideal situation, no one will notice, and nothing will go wrong.

But your team knows that things do go wrong, and they go wrong often. The team enjoys having safety nets because they're free to try things and get

them wrong; someone will catch problems before they reach the client. Removing these precautions has the effect of making your team both more cautious and less confident. This is the worst intersection for your team: they're working more slowly because they need to compensate for the lack of double-checking, and they don't feel as sure of what they're doing.

At one of my past gigs, we had a brilliant performance engineer named Tim. The phrase "Tim tested it" was the equivalent of "We're certain of its performance." Whenever a project wasn't working as expected, someone could expect to hear, "Did Tim test this?" Project managers who answered "No, there wasn't enough time" immediately realized they were saying the wrong thing, because they were actually saying, "We were running late, so I took a risk and skipped that part. I hoped it would be fine. Looks like it isn't."

Adding risk to a project is sometimes the right thing to do. But you must treat it the same way you treat other risks to your project. Make the choice intentionally and transparently. If you decide to pursue a "speed up" strategy that will make you and your team less comfortable about the quality of what you're delivering, make sure your client or program manager knows about it. There is little value in keeping it to yourself, no matter what you think at the time. Present it as an option, just as you would anything else. Let your program manager or the client have a chance to talk you out of it or accept it as a risk and move on. Don't keep it a secret.

Summary

As I said at the beginning, this wasn't intended to be an exhaustive list of all possible expected problems. I could probably fill another book and call it *Things Project Managers Consider Bad Luck That They Should Have Seen Coming*. That said, this chapter has shown you samples of the types of things your client and program manager see all the time across multiple projects.

These foreseeable problems span all facets of the project, from client management and participation, to team management, to how you decide to deliver. Every part of a project has elements that can go wrong and have gone wrong many times in other projects, and your program manager has probably seen all of them. In fact, program managers are generally more excited if you bring them problems they haven't seen before, because at least they're dealing with something new.

When you're ready to bring problems to your client or your program manager, ask yourself if you could have foreseen them. Then ask yourself if you did anything to cause them or make them worse. Many project issues fall in

one or both categories. This is okay—these are common problems for a reason. But you need enough awareness to not call them bad luck, “perfect storms,” or unexpected problems. There is no better way to lose the faith of your team or management than being unable to expect the expected.

Establish Metrics

“Who is keeping score around here?”

As in many organizations, a project manager on my team is responsible for more than just delivery of a set of requirements that they or someone else may have gathered and documented. In this case, the project manager is responsible from the time someone has a bright idea about something we ought to be doing, to the time it's delivered and creating value for the company. This opens the window quite a bit from the narrow responsibilities of simply delivering on the goals of the project. Instead, project managers are involved all the way from beginning and discovery to the end of deployment and transition planning.

Because of the broad nature of the role, it's difficult to come up with a simple mechanism to determine whether the project manager is doing a good job. A project manager who is excellent at conceptualizing may fall short when it comes to operationalizing. One who is great at planning and scheduling may be poor at risk mitigation. Or you may be a generalist—superlative at nothing, but competent in all facets of project management.

This is a world of metrics, where everything is quantified and measured, and financial ratios are applied, such as return on investment (ROI), return on equity, and other things of that nature. As a project manager, you're probably not being held to financial metrics like those, but you need a way to know whether you're doing a good job.

Complicating this problem is that every project is different and can have different goals. Executing on a well-known project plan, one that you've done before, should have a much tighter expectation for delivery metrics.

Managing a project the likes of which no one in the organization has previously attempted may have a very different set of goals and success criteria.

■ **Note** Lack of clarity around project metrics isn't a reason to ignore them entirely.

That being said, although it's okay to adjust the values you hope to achieve in the metric you're tracking against, it isn't okay to avoid having metrics in the first place. If you don't measure how you're doing, then you gain no knowledge for the next time you do something similar, and a lot of organizational learning is gone.

Top-Level Metrics

Projects usually have one or two top-level metrics that you're tracking toward. These are usually profitability, ROI, customer satisfaction, or something similar. From a business perspective, whatever mechanisms you use to achieve the goal, even if not best practice (or even good practice), are okay because they further the goals of the project. Short of doing anything unethical or illegal, a very poorly run project can still produce spectacular results.

But that doesn't help you become a better project manager. First, you probably don't have much influence over your projects' profitability. You can do everything correctly, and salespeople can sell your products at a loss, or the market may receive the offering poorly, or it may be shelved shortly after launch due to a change in priorities. Second, doing whatever it takes to deliver does nothing to help you grow in the practice of project management. You're interested in knowing how you did as project manager, with the items outside your sphere of influence removed. To improve your management technique, you must know what you're currently doing and find a non-anecdotal way to determine if you're improving.

Because project management is multifaceted and heavily nuanced, it's difficult to find any single metric that lets you know how you're performing. And because maximizing one part of project management doesn't tell the entire story, you're faced with using several metrics at the same time to help build the complete picture of the project's management. Such metrics are called *indicators* because all they can do is give you an indication of how well the project is being run. Taken collectively, however, they can spin a powerful and understandable tale.

There are dozens of indicators you can use, but relatively few provide probative information about the project's performance. These metrics are therefore called *Key Performance Indicators* (KPIs). You can use these KPIs, in part, to help give visibility into your projects. It's vital for your team, your client or sponsor, and your management to have a clear picture of how a project is going. Without this visibility, you wind up with anxiety, or you're stuck using anecdotes to determine your current status. And that kind of confusion and lack of clarity eventually causes drama in your projects and disruption in your project team.

What Are KPIs?

At their heart, KPIs for project management are measurements that are quantifiable and can be used to measure a project's health. They can provide information and visibility into the project, and they can be used to make decisions without seeing what the project is actually delivering. If a project is going off course or needs changes in structure or management, waiting until the delivery date is often too late to save the project. Better to measure along the way and track how you're doing, in order to make adjustments.

Often, project KPIs are measured and reported on a regular basis. This is done either passively, during the project's standard update, or actively, during something like a project audit. They're used either to instill confidence in the client or sponsor, or as a way to prove that things are going right—or going wrong. Both messages are important for a project manager to have available, and data that supports either side is helpful in many situations.

Many things make a KPI a good one to track. However, a good KPI must have at least three traits. It should:

- Align with the goals of the organization
- Drive the right behavior
- Be based on valid data that can be proven

Without these characteristics, KPIs wind up doing more harm than good. They can cause the wrong behavior, obfuscate reality, or lead a project astray because the wrong thing is being measured. Let's look at these three traits in more detail.

Align with the Goals of the Organization

All organizations have goals. I worked for a company whose stated goal was to disrupt and dominate an entire industry. In fact, "Crush the Competition"

was often specifically stated among the company's goals, products, projects, and efforts. Our goals were not things like "Provide an easy lifestyle for our employees" or "Give lots of money to worthy charities." Although we did our fair share of the latter, the former was specifically not in the charter of any manager.

Every now and again, a business owner or project manager would include items in their project that didn't have to do with profitability or increasing market share. Whenever this happened, I'd have to say to them, half-jokingly, "Are you new here?" It would certainly be nice to give our customers a wonderful experience, or figure out how to provide a better product at the same cost, or do some other objectively "good" thing, but that simply wasn't what the organization was all about.

■ **Note** The KPIs need to match the organization, not the other way around.

Not surprisingly, the company was laser-focused on ROI. Because the goal was to dominate the industry and crush the competition, making sure everything we did had a proper level of ROI was crucial to project and business success. Project managers could do little to influence the *R* part of the equation, so their main impact was on making sure the *I* part was well known and well tracked. Making sure the KPIs being tracked were aligned with the goals of the company were critical to their usefulness.

If you find yourself running a project where the goals aren't aligned with the company, you should be prepared for disruption. Obviously, if they're completely misaligned, then you have a whole set of problems to deal with, but this is rare. More likely, you're running some projects that include good elements, but other projects include "nice to have" items. In the organization I mentioned, someone was running a project to donate our "imperfect" products to charities. The original project statement was wonderfully philanthropic and talked about fairness, enriching lives, and doing well. What was missing from the statement was "maximizing tax benefits."

I tried to get that written in as a high-level goal, but I wasn't successful. Not surprisingly, before the project finished, we decided to give the imperfect products to the nearest charity that was willing to come pick the stuff up from us. The team was then reassigned to do something that would increase corporate profits. Can't say I was surprised.

Drive the Right Behavior

There is a statement in management that you “get what you measure.” If you want to measure employees based on how many phone calls they can make in an hour, you can bet they will make those calls as fast as possible. They may not generate sales with those calls, but they will make a lot of them. This is an example of an unhelpful KPI.

Another example comes when you give conflicting goals to different groups. For instance, you may give the sales group the goal of raising the average value of a customer and while the production group the goal of lowering costs. While the sales group is out trying to sell premium upgrades and extolling the virtues of the product’s quality, the production team is looking for ways to shave a little off each corner in order to meet their goals. Best case, someone from sales calls the manufacturing team and convinces them to find cost savings elsewhere. Worst case, you wind up with customers who pay for a high-end product, but receive the cost-cut version, instead.

■ **Note** It’s not uncommon for two groups to maximize their own performance while harming the company or project performance.

It’s simple human nature to try to maximize performance while minimizing effort (what I have called *personal ROI* elsewhere). This isn’t incorrect for employees to do. By giving them something on which they will be measured, you explicitly tell them what you care about. So, employees are completely correct in trying to do their best against those metrics, even if they don’t do much to create success for the company or the overall effort.

Your measurements should do the following things:

- Drive the behavior you want.
- Optimize the success of the project or organization.
- Help with making decisions.

I have already discussed the importance and pitfalls of making sure you’re creating and supporting the right behavior. The other two are equally important.

Metrics that measure an individual’s performance create the desired performance from that individual, even if that’s not the right thing to do. The employee who is incented on making lots of phone calls may not spend time

teaching the person next to them how to make calls, because that will cost the employee 20 minutes and maybe keep them from reaching their goal. Also, if used individually, such a metric can be used as a way to blame someone for “not pulling their weight” when they were doing the right thing by the organization or the company.

As with anything else, measurements need to be seen from a portfolio view. You have a group of people, a group of metrics, and a window of time. If you try to narrow this portfolio to a single instant in time, a single metric, or a certain individual, then you’ll be misled. It’s important to note that you should be trying to maximize the value across your entire project over time, and do the measuring that way.

Finally, a useful metric helps with making decisions. It’s not helpful to measure things that people don’t have to decide to do. For instance, the metric “Curse at the customer less than once per day” isn’t all that necessary—you don’t want anyone cursing your customers at any time. But more important, it’s very easy to have metrics that don’t change anyone’s behavior, either positively or negatively. These aren’t necessarily bad metrics; they just do nothing.

For instance, at a past gig, we had a metric for “days between requirements finalization and sign off.” I suppose this was a fine thing to check. We didn’t want our requirements lingering too long without being settled. But so what? No value was created when a project manager got this time down to hours or minutes, rather than days or weeks. No decisions were made differently based on when the documents were signed and completed. The only behavior this metric drove was the level of harassment the project manager felt empowered to give the people with signature authority, which they didn’t appreciate.

When choosing what you measure, you need to determine what behavior will change based on the metric, whether the metric maximizes the correct thing, and whether the existence of the metric will help alter decision making. If what you’re choosing to measure doesn’t do those three things, then you may have a metric that feels good but doesn’t create any value. And that’s if you’re lucky. If you aren’t careful, you may create a measurement that fosters the wrong behavior or optimizes for the wrong thing entirely.

Be Based on Valid Data That Can Be Proven

The first two criteria—making sure your KPIs are aligned with the organization, and making sure they’re changing behavior in ways that you want—are both strategic requirements for a good metric. But a metric can pass both

tests but still fail if you don't have good data to measure it accurately, and in a way that everyone believes. If the people or projects being measured feel as though their metrics are more subjective than objective, then they will discount them entirely.

For instance, I once consulted for a company that had this goal: "One million satisfied customers." As a vision, it worked. It was a big round number and a result we all could hope to accomplish. Shortly after the announcement of this new vision, "thermometer" tracking posters started showing up around the office. People stopped viewing the lofty, nonspecific goal as a vision and began viewing it as a metric. And as soon as that happened, disputes started.

This four-word phrase was so ambiguous that it was practically useless. By *one million*, did the company mean households or unique customers (do a husband and wife count as two)? One million customers ever, or that have purchased in the past year? What is a *customer*, anyway—someone who bought something? How about someone who bought something but then returned it? How about someone who buys every week? Should they be counted more than once? And what does *satisfied* mean?

The company did manage to hit this goal, but it was much more of a morale boost than an actual financial event—not that there is anything wrong with that. As something for the company to measure and march toward, it was terribly confusing. A good metric must be clearly understood, must be determined by something akin to a mathematical formula, and must include a value for the goal.

■ **Note** When it comes to metrics, math is good.

I worked for a company that sold a product that was purchased as a subscription with a recurring monthly fee. One of our biggest problems was *subscription churn*: the number of subscribers who canceled or didn't renew when their subscription expired. But even that definition wasn't precise enough. We decided to define churn as follows:

The number of subscribers we have now, that we also had twelve months ago, who haven't gone out of business, merged with another company, or discovered a conflict of interest, and haven't canceled, downgraded or otherwise stopped paying for their subscription, divided by the number of subscribers we had twelve months ago.

This turned out to be the most complete definition of *churn* we had ever compiled, and it's the one the company still uses today. It was specific and unambiguous enough for us to measure against, and we had a huge amount of very obvious, very valid data to use. (Did the customer send a check or not?) However, it also left a little wiggle room where healthy debate could happen, leaving the team in a position where specific situations could be discussed.

But it wasn't enough to merely come up with what we were measuring. We needed to determine our historical measurement of this metric, and we needed a forward-looking goal. What was interesting was that the company had always thought it had a roughly 25% churn rate, based on billings and what people knew about our accounts. This was frustrating: one in four customers decided to cancel or otherwise drop after a year. When this more precise metric was applied, the churn rate turned out to be closer to 20%, moving it from one-in-four to one-in-five, which immediately made a bunch of people happy.

■ **Note** Sometimes measuring something in a new way provides valuable insight.

The company set a very ambitious goal of cutting this number in half, but there was a strong recognition that this would be a team effort, and no one project could be responsible for the overall goal. We were able to give specific targets to certain products and projects, such as reducing churn by 50 basis points for subscribers to a specific product or those who had used a specific feature. It was a great way to measure the value of a project while still aligning very tightly with the goal of the entire organization.

Making sure what you're measuring is actually measurable in a way that is probably true is a very important aspect of a good metric. Without a repeatable, almost auditable way to validate how you're doing, your metric will be disputed, distrusted, and eventually, disused.

Some Example Project-Management Metrics

You now know why metrics are important, and you know what makes a good one. But it would be a huge waste of time and effort to re-create new metrics for every situation. From afar, many projects look the same: a bunch

of experts working together to reach a particular outcome, with a project manager at the helm. The team changes, the project manager changes, and the challenges change, but at heart, projects have a lot of overlap. Because of that, many projects reuse metrics from other projects or other organizationally well-known goals.

There is also a side benefit to using metrics that are well known and well understood. For a metric to be useful, it must resonate with the people on the team. There are few ways to bring drama into a team faster than telling members you're going to measure them on something complicated, or something they have never heard before. It's even worse for project managers. You know how metrics work, and you're very comfortable with them. But if you're given one that makes no sense or doesn't apply in your situation, then you'll either ignore it or worry about it. Both of those situations cause you anxiety and stress, and your team recognizes it.

With that in mind, project and program managers can use two well-known metrics to measure in-flight project performance. They have to do with how well you're using the resources and people that were given to you as part of the project, and how well you're tracking toward delivery. The former is generally tracked fairly simply and can be done with normal project artifacts such as time cards and expense reports. The latter can be determined using the concept of *earned value*, which I describe shortly.

Resource Utilization

Depending on your organization, *resources* can mean different things. It can be people such as software engineers or consultants, it can mean money or budget, it can mean time in the lab, or it can mean the use of some fairly scarce piece of equipment. Whatever it turns out to be, at the start of your project, a predetermined bit of this resource was set aside for your use. The question becomes, did you use it? And did you use it wisely?

The second of these two questions makes for a very poor KPI. "Using resources wisely" is hard to measure objectively in any agreed-upon manner. Sure, if you took the entire project budget and bought lottery tickets with it, a case could be made for unwise use of resources; but short of that, although hindsight may disagree with your choices, it's difficult to know if you are using your resources wisely before your project is delivered.

The first question is easy, however. It's a pretty straight calculation. Of the resources you were given, how much did you use? To be fair, there is nuance involved. Resources are granted ahead of time after some analysis of

what the project requires. After the project begins and is better understood, it may turn out that some of the resources on the project are no longer needed. This isn't unusual. In my organization, project managers are obligated to release resources back to the available pool as early as they're able. There is even a metric for this. If you're scheduled to use a resource but decide three or more weeks ahead of time that you aren't going to use it, then you can release it and have it removed from your project. If you wait too long, then even if you don't use the resource, it becomes attached to your budget.

I sometimes have a difficult time explaining this to my team. Surely, coming in under budget must be a good thing, they say. Don't get me wrong—it's good not to go over budget. But there is a cost associated with resource utilization that is hidden from project managers: opportunity cost. Let's use a simple example and say you're talking about money. At the start of your project, you ask for \$100,000. Halfway through, you realize that you probably only need \$50,000, but you hang onto the extra, just to be sure. The total comes to, let's say, \$55,000, and at the end of the project you return \$45,000 with a smile. But as portfolio manager, I won't be smiling back.

■ **Note** Opportunity cost is often more expensive than the cost itself.

What you aren't appreciating is that the \$45,000 could have been used elsewhere. Another existing project could have used it, or I could have funded a new project—and maybe even hired someone to run it. So although yes, you're returning \$45,000, those are last year's dollars. I should have had a team turning them into this year's dollars.

Resource utilization involves being a wise steward of a project's budget. Don't be over, and don't be too far under (without releasing your hold): in the former situation you're spending more than you have, and in the latter you're wasting time. This is what makes resource utilization such a good metric to follow. It's easy to understand, because resources are usually measured in hours or dollars, and it's not difficult to track. Additionally, this is one of the things most in the project manager's control. Between proper scoping and proper planning, it should be a fairly straightforward exercise to stay within tolerance. To put it the opposite way, not staying within tolerance is almost always a failure in project management.

Earned Value

One of my favorite methods of tracking the performance of a project is *earned value*. An earned-value chart tracks three things simultaneously and plots them against each other: scope, budget, and time (as created in the initial project plan). *Scope* refers to an objective measure of value creation, such as function points, features, or use cases completed. It's meant to show how much of the project is complete.

This value is then plotted against both time and money. You wind up knowing how deep you are into the value the project is supposed to deliver, and how much time and budget remain. If you've delivered 10% of the project's value but have used 90% of the scheduled time, that might be a problem. Also, because it's a chart, you can track the slope of the line and determine whether your project is slowing down, speeding up, or has plateaued.

If you're familiar with a burndown chart, it's somewhat similar, although with a very different focus. Burndown charts measure how much work remains and show how much work is left to do in order to deliver on all the original requirements of the project. The focus of an earned-value chart is to show the client or customer how much value has been created (or earned) by the project to date. Showing a burndown to your clients reminds them of how much work is left *not* done, rather than how much has been accomplished.

As a program manager, I can learn a lot about a project and its manager from this chart. Because you're plotting actual value creation versus planned, showing the budget, and laying it over time, you can quickly determine whether a project is in the right ballpark of being on track. Some deviations are to be expected: sometimes a project gets ahead, sometimes it falls behind, but this isn't alarming. This chart also shows direction; if the project was behind two cycles ago and now is even more behind, that may indeed be alarming. Or the reverse—it was behind two cycles ago and has now closed some of that gap. Such a project may be headed in the right direction. For such a simple graph, it's amazingly powerful.

Figure 12-1 shows what I mean. Consider a project that is scheduled to run for ten cycles and that is currently finishing cycle six. The Planned Value line represents the original project plan: you were going to hit 100,000 units of value (say, function points) by the end of cycle ten. Therefore, to be directly on target, you'd be halfway done by the end of cycle five. The Earned Value line represents actual performance. As you can see, you started out a little ahead but fell behind around cycle four and now seem to be losing ground. If you extrapolate that line, you see that without some changes, the project will

disappoint. And although it does seem that costs are tracking lower than expected, cost is trending up and may shortly surpass the actual value created.

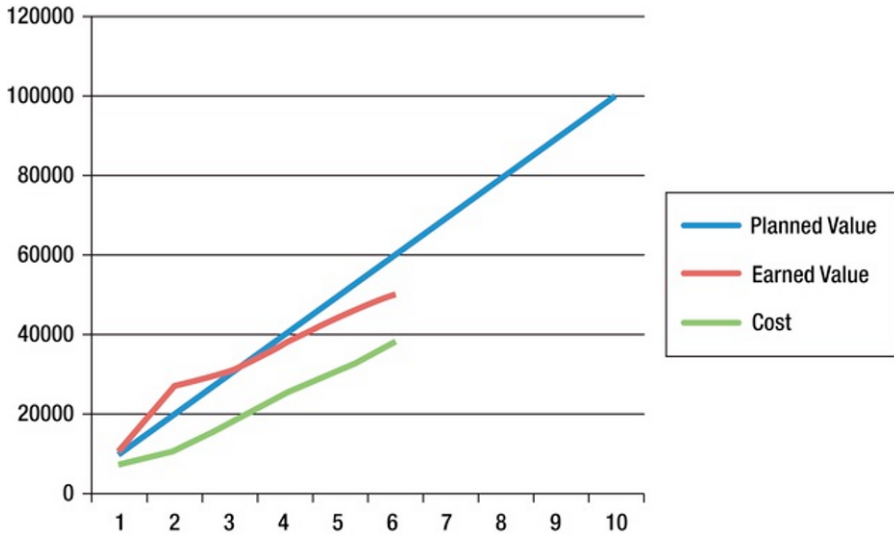


Figure 12-1. An earned-value chart

For a single view, this chart is a wonderful picture of a project's status. It does have a few weaknesses, however. First, it's helpful only for a finite, discrete project. If you have an ongoing effort or a project that doesn't have good completeness criteria, then the Planned Value line is always in motion, which defeats the purpose of the graph. Additionally, although this technique does have a notion of things being "complete," it doesn't ensure that delivery is "good." Quality of delivery needs to be tracked separately.

Of course, project managers track many other metrics, such as schedule and cost variance, which essentially mean how close you are to the original plan. Project managers also track defect rate, milestone completion, and tollgates that have been passed through. In general, however, I look at the resource-utilization and earned-value KPIs to get a sense of how a project is doing while it's running. Depending on your situation, there are other things to keep in mind; but even with just these two, you can get by fine.

The Two KPIs Your Program Manager Cares About

In medicine, there is an old chestnut that goes something like, “The operation was a success, but the patient died.” This somewhat flippant statement contains quite a bit of truth. Every indicator along the way told you that you were doing well, but the final result wasn’t what you hoped for. In other words, you hit all your KPIs, but the project itself was a failure.

Realize that your program manager cares about the project KPIs and will ask about them and track them. But managing an active project is the job of the project manager, not the program manager. That’s why you were hired in the first place. At the end of the project, the program manager cares about only one thing concerning you, the project manager: what your prospects are for running a project again. And for that question, there are two audiences: the client and the team.

The client has many meanings. It can be the literal client, the executives involved, or the end user of the project. *The team* usually means the project team: people who self-identify as being on the project. But it can also mean the people who are on the extended team, or other interested parties to what you’re building. But the question is the same: do they want you managing their project next time? Let’s take a deeper look.

Client Satisfaction

In this context, *client satisfaction* doesn’t mean clients are happy with the result they got or whatever the project was supposed to deliver. Nor does it mean whether the project added to the client’s success, was profitable, or met any of the objective measures of process success. I usually phrase this as follows: “Would you want this project manager on your next project?” This single question bakes in an awful lot of topics.

Several aspects go into client satisfaction. Among them are quality and speed of delivery, collaboration ability, judgment and decision making, and level of expertise both in the problem domain and in project-management best practices. Every client has a few of their own, as well. Virtually everything on which a client can measure, judge, and evaluate a project manager can be wrapped up in that singular question.

Answers to this question cover a wide range:

- The project didn’t go as well as we hoped, but that wasn’t the fault of the project manager. I’d happily have them on my next project.

- The project suffered from poor communication but was otherwise well run. I'd take the project manager back, but only if someone can work with them on their communication and messaging skills.
- I'm sure the project manager knows what they're doing, but I think they're wrong for the next project I have in mind.
- Please, no. Anyone else would be fine.

Of course, there are lots of steps in between. But I have had a lot of success with this specific question, because it puts the evaluation in the real world. All projects are difficult to run, and they all have issues; this isn't a surprise. A client may be satisfied that they got as much as they could out of the team and the situation, but the client may feel they should have gotten more. Maybe they got the exact delivery they expected, but they didn't enjoy working with the project manager. Or maybe they're unwilling to question the competence of another professional, but they can give the program manager the message by being firm that they don't want to work with that person again.

This is a much more powerful question than the normal end-of-project surveys that are sometimes used to measure performance. It's possible to give the client exactly what they wanted and have them be dissatisfied. Likewise, it's possible to disappoint the client but do so with such client-management skill that they would be happy to give you another try. Keep this question in mind during the project, and know that most effective portfolio managers will ask it when the project is over.

Team Satisfaction

The counter-balance to how your client feels involves finding out how your team feels about your management. Even if the client wants you back, if the team you were working with won't work with you again, then you still have a problem. And again, the question is the same: "Would you like to work with this project manager again?" Much as with the client, the question can reveal a lot.

Team members have lots of different reasons why they may have enjoyed working on your project. Maybe you were particularly hands-on or hands-off. Perhaps you have incredible skill at documentation and written communication. It could be that your meetings were well run and highly efficient, or that you did an excellent job fostering a good team atmosphere, one they'd like to participate in again.

The opposite is also possible. You may have held too many meetings that were a waste of time. Possibly you had no expertise in the area of the project and were unable to add much value to discussions, even as a facilitator. (Or worse, you had no expertise in the area but were participating well above the level you should have.) The reasons a team member wouldn't want to work with a project manager are varied and wide ranging—and not all of them are valid. You're looking for an aggregate opinion and trends.

Project management in an organization is a balancing act. You can't always favor the client at the expense of the team, nor can you favor the team at the expense of the client. Both sides know this and are willing to be flexible, given the situation and a certain degree of fairness. The judgment call on which side takes the brunt of a decision is one that even very senior project managers struggle with. But it's the kind of thing that you need to work work on.

Summary

Project management is at least as much an art as it is a science. Because of this, project managers are sometimes resistant to put measurements, or KPIs, on their performance. This is almost always a mistake. Without some way to tell whether a project manager is doing a good job, a program manager can't know what is happening during a project or how to evaluate the project manager after the project has finished.

To be worthwhile items to measure, however, the KPIs you use must be aligned with the goals of the organization, foster the kind of behavior and performance you want, and be based on valid and verifiable data. If you create metrics that don't reach your organizational objectives, or that create (or even incent) the wrong behavior, then your KPIs are leading you in the wrong direction. Additionally, if you create ones that can't be proven or are very subjective, then they will be discounted and disbelieved by many.

The two measurements I find useful to evaluate in the middle of a project are resource utilization and earned value. The former is a measurement of the project manager's judgment of how to use budgeted resources, which may be people, money, or a limited supply of something. Knowing that resources are being used as planned is very important to a program or portfolio manager. The latter is much more of a way to track project delivery. The two put together give a good, although incomplete, picture of a project's status.

When the project is over, two questions need to be answered. Would the client like to work with you again? And would the team like to work with you again? Both are vital pieces of information as your program manager decides what your next assignment should be, or whether there should be a next assignment at all. The balancing act between advocating for the client and supporting the team is difficult, but it's important for your long-term career and place within the organization.

Know the Roles

“Hey coach, tell me again what position I’m playing?”

The way the project manager structures a team is perhaps the single most important factor in determining how the team will function going forward. Do you want a balance of skills and expertise, or do you want to make the team heavy in one aspect? How much interaction with the customer or business owner do you want the team to have? Do you expect that members of the team to work together constantly, or only at the edges? These decisions and many more largely determine what kind of project you run and how it operates.

Additionally, each person on the team needs to understand their role. Chapter I discussed the roles in the overall project, including project manager, program manager, sponsor, and others. That chapter briefly mentioned a role called *team member* and talked about the team members’ responsibilities to the client and to the project. What I didn’t mention was their responsibility to each other and what they need to do to truly become a project team. That’s what we will explore here.

Usually, the project manager is responsible for the organization of the project team. Sometimes the project manager is handed a group of team members. Other times they’re told to recruit or develop them. However the project manager acquires a team, they need to make the most of it. And no single solution work for every project or every team.

Depending on how much history the project manager has with the organization or client, there are different strategies for creating a team. For instance, I've had project managers who develop talented team members and then bring them along from project to project, because they know and trust those members' abilities. I've had other project managers who were indifferent to history and instead always chose or recruited the best person for the job that needed doing. I've seen both strategies work, and I've seen both strategies fail. There is no one answer.

■ **Note** Building a team is much more art than science; there is no one right answer.

What I have seen fail often is a project organization that doesn't do this kind of resourcing intentionally. Recently, my portfolio included a project that I believe was staffed with what I will charitably call "leftovers." These were the men and woman no one else wanted on their projects. None of them individually was a problem; in fact, many successful projects have run with them as team members. But I had never seen a successful team made up exclusively of these kinds of people. I told the project manager that I would be willing to either restaff the project or cancel it. He declined both options. About a month later, he was regretting it.

What Makes a Project Team?

Before a project team can be forged and structured, it's imperative that the project manager understand what makes a project team in the first place. Project teams are slightly different than standing teams, functional teams, or many other kinds of teams. Because the team is brought together for a specific purpose, for a finite amount of time, many of the considerations that other teams need to take into account are less important, and a few others are more important.

When you're thinking about your project team, consider that it's a group of professionals who are responsible for performing project tasks, both as a team and individually. This group helps to achieve project goals and objectives for the primary purpose of reaching a small set of success criteria as set out by the owner of the project.

Things Team Members Must Know and Do

Everyone on your team needs to be able to handle the following:

- Understand the needs and objectives of the project.
- Execute on project tasks.
- Express status.

Without a common ability to perform these items, then your team will never reach a high-performing stage. If you have team members who can't fulfill these needs, then you need to either locate suitable replacements find a different role for those individuals to play in the project.

Understanding the Needs and Objectives of the Project

I nearly said “business” or “client” here, instead of “project.” But on reflection, I think knowing the objectives of the project are enough. It would be fantastic if every team member had a grasp of the business need, how the client will use the outcome of the project to be successful, and how the project itself helps contribute to that success. But that may be too high a bar for every team member to reach.

■ **Note** It's not important for team members to internalize the goals of the client, but they should be able to understand the goals of the project.

It may be good enough if all team members know the project goals and understand how their part helps reach those goals. They may not need to know how what they're working on will eventually contribute to the success of the client, but they must understand how what they're working on will add to the project's success. Without this knowledge—and the interest to gain this knowledge—it's very difficult for a team member to remain in alignment with the goals of the project.

Executing on Project Tasks

Many pieces make up proper execution. They include planning tasks for yourself as well as others, estimating how long items will take to deliver, and

then delivering them. You may think these are necessary skills that anyone would have, but that isn't an assumption a project manager should make.

Although I'd like to be able to say that only the most junior people need help in this area, experience teaches that this isn't true. The ability to create a plan and then deliver on that plan isn't a skill many people are born with. I've worked with many otherwise brilliant, very senior people who couldn't plan their own workday, let alone the delivery of a complex task. This is okay, as long as you recognize it from the start and are willing to work with the person to make sure they're on the right track. If you don't have the time or inclination to do so, you may need someone else on your team.

Expressing Status

Similar to being able to execute on their own tasks, your team members need to be able to communicate how they're doing, either to the project manager or to each other. This skill is similar to the ability to plan your own work. If you can understand how much work needs to be done, then you should be able to express how much is left to do. But even when you can't do this, your team members should be able to say they're finished, or not started, or more than half done, or less than half complete.

Complicating matters is the idea of *effort time* versus *actual time*. Developers have said to me, in complete honesty and sincerity, "I need ten more hours to complete this task. It will take me six months before I can get ten free hours." Many team members aren't in control (or necessarily consulted) about their assignments. This leads them to say that they only need a day or two to get something done; but when a day or two, or a week, passes, the task still isn't finished. They look at you, puzzled, and continue to say that they only need a day or two—and if you, the project manager, can help them find that day or two, then they can finish the task. If only it were actually that easy.

These three skills are essentially baseline team-member skills. The first is being able to and possessing a willingness to understand the objectives of the project and make sure that whatever you're delivering contributes to meeting those objectives. The second is being able to do the work that is assigned, including planning and delivering. And the third is being able to explain how you're doing in a way that the project manager or client can digest and comprehend. If team members don't seem able to fulfill all three of these requirements, you should consider dismissing them from the team or figure out how to properly manage them without doing their work for them.

The Three Levels of Team Members

If you're familiar with the parable of the chicken and the pig,¹ what follows shouldn't be a surprise. Essentially, a chicken and a pig decide to get together to make breakfast. The chicken is able to make its contribution—some eggs—and then move on to the next task. The pig, however, needs to be totally committed to making this breakfast, because in order to contribute some bacon, it has to leave a piece of itself behind.

There are three kinds of team members: leaders, members, and contributors. Virtually every project includes people in all three roles, and large projects have multiple people in each role. Each person should self-identify into one of those roles. If people's opinions of their roles don't match what you believe, then you'll eventually have a conflict and a problem to resolve.

Leaders

There should be relatively few leaders within your project. This doesn't mean you should have few people with leadership skills, characteristics, or instincts. But projects can only survive if a limited number of people—you and a few others—are yelling "Follow me!" The smaller the project, the fewer leaders it should include.

Leaders are people who provide guidance to the team and are accountable for the delivery of the project tasks. The role includes development and skills training of project team members, as well as understanding their motivations and personal drivers for completing the project. In some cases, leaders can take a very positive stance and gently guide the team in the direction they want to go. In other cases, the leaders need to take a firmer position and direct people specifically. Either way, a project can't survive without someone leading it.

■ **Note** Leadership is an important role in a project team. Your project most likely can't survive without a leader.

¹ http://en.wikipedia.org/wiki/The_Chicken_and_the_Pig

The leaders on the project should be recognizable by everyone inside and outside the team. Depending on the size of the project, there can be one leader or many leaders. Sometimes the project manager takes on the role of client management, development manager, and deployment manager. In larger projects, multiple people may play each role, each with a slightly different nuance. The important point is that every project needs at least one person whom everyone knows to look to when they need help, guidance, or direction.

Team Members

Team members are people who would self-identify as being on a project: if someone asked them what team they were on, they would list your project among the things they were doing. This is a key distinction that you should consider at different points in the project. These people are directly accountable to the project (not just a single task or a handful) and are actively managing its delivery or scope.

In some cases, project managers call these folks their *core team*. They're with the project for a long time; they have an interest in its success, if not an actual stake; and they expect to get something out of the delivery of the final result. They're also looking to the project for other reasons, such as career growth or skills development, or they're motivated by the project's expected outcome.

I understand the notion of calling these people *core*, but it's not a good way to think about them. The people in this category *are* your team, and taking care of them and their delivery for the life of the project is your job.

Team Contributors

In contrast to team members, team contributors deliver on a series of tasks for the project but aren't accountable for the project's success—merely the delivery of their specific tasks. These are your chickens from the earlier parable. Projects can't survive without these types of people: they set up your servers, or get you access to something you need, or have a very specialized skill that you occasionally require.

Team contributors take tasks from the project leaders but take direction from their functional or permanent home. No matter what pressures a project is under, contributors follow the proper procedures to deliver on their tasks, even if the decisions aren't exactly right for the project. This is correct

behavior for them. They're likely to be responsible for the health of the project's infrastructure long after you've moved on. Convincing them to cut corners or steps in their process may help in the short term, but at a long-term cost.

■ **Note** Project contributors are often more concerned about the long-term health of your deliverable than you are.

A project manager must understand the difference between contributors and team members. I once had a project manager running a fairly simple project that required a novel use of clustering in our web farm. This task was scoped to be about ten hours of work, so a network IT guy was slated on the project for those ten hours. At the end of the first month, he had booked five hours to the project. When we went over the time cards, I casually asked the project manager if that meant the IT guy was about half finished.

The project manager said, "Oh, no, he hasn't started yet." Puzzled, I asked why he had booked five hours to the project. I was concerned that if he hadn't started yet but had used up half his allocation, he might not finish the work. The project manager looked at me in all seriousness, and said, "Oh, that's probably just the weekly meeting. You can ignore that."

As you can imagine, this didn't work for me. Because the project manager was treating a contributor like a team member, we had burned through half the available hours, and the contributor hadn't produced anything. When I asked the project manager what he was thinking, the answer was, unfortunately, reasonable: "Well, I thought he'd want context on what the project was doing and why we needed his help." This was a great example of an answer that is hard for me to explain.

Yes, it's helpful and maybe good for the team if your network IT guy understands the project. If he internalizes the project's goals and objectives, perhaps he can complete the work he needs to do in nine hours instead of ten and make suggestions that might help you in the future. But that comes at a cost. For your team members, you're willing to pay that cost—in fact, you *want* to pay that cost. On the other hand, if people are only contributors, you've explicitly decided ahead of time that you can't use them this way. Understanding the difference is vital to proper execution.

When you organize the team in an easy-to-understand and appropriate way, you have people in all three roles. Some are leaders, some are team members committed to the project, and some are expert contributors who are accountable to their own function but willing to do whatever work you require. Good collaboration requires that everyone associated with the project knows which of these they are and can conform to meet the expectation of these roles. It's your job as project manager to explain to the project's members what their roles are and to manage them accordingly.

The Importance of Role Definition

Defining the roles within your project is the first step toward making your group of people into an actual team. And it's this teamwork that you want to foster. People who identify as being on the same team are predisposed to like each other and work with each other, and they start from the default position of everyone wanting the same thing. Beginning from that stance makes a lot of problems disappear; and problems that exist are easier to solve better because people are working together to help reach the same objectives.

Building your team to be effective and high performing is one of the more important tasks of a project manager early in the project. Structuring the team for success greatly influences the way that the team operates and works together. Creating a team whose members collaborate well, share their skills and knowledge, and support each other in pursuit of the project goals goes a long way toward a successful project. This requires correct and appropriate use of members at all levels, and clarity among the team members about their roles on the project. Creating that structure and that clarity is the job of the project manager.

■ **Note** Structuring your team properly is key to creating a successful team.

A high-performing team doesn't just get a lot of work done. It's also important that the team provide a wide array of perspectives and ideas, potential solutions, and implementation strategies. These include everything from advocating for the users to basic technical best practices, as well as paying attention to things like performance, efficiency, and ongoing support costs. Unless people know where they are in the project, you may create a collaboration challenge without realizing it.

In a high-performing team, people in every role, at every level of engagement, know where they exist in the project organizational chart. This may not be a literal chart that you tack up on the wall, but it should be in people's heads. If people are uncertain about their role or not completely sure whether they're on the team, then you have a problem to solve. You're much better off being up front about your expectations and the needs of the team roles than trying to correct behavior after the fact.

Project Team Member Roles

Within a project team, members play different but equally important roles: functional manager, analyst, developer, tester/reviewer, and client interface. If your project is very small, team members may be asked to play more than one role at the same time. If your project is large enough, several people may play each position, just with a different focus. No matter what size team you wind up with, each project manager must decide who is playing the following roles and how they interact with the project itself.

The list of roles isn't balanced. It's not unusual to have 1 person playing 1 role, and 40 playing another. Nor is it unusual to have a team of three where each member is filling two or more roles at the same time. The key is to make sure all roles are covered and available to the project. Without them, your project is missing a key component that you must figure out how to get elsewhere.

Functional Manager

It's rare for a true functional manager (the manager of a reasonable-sized team) to be a full member of your project team. If they're dedicated to your project, then who is managing their team? However, there are cases where entire teams are assigned to projects, so you get the manager as a part of the bargain. When that occurs, you should treat the manager as a reviewer, which I get into in a minute.

I'm talking about the functional manager who is most closely affiliated with your project. If you're installing a new accounting package, it could mean the accounting manager; if you're installing a new HRIS package, it could be the IT manager for the HR department. This kind of member straddles the line between stakeholder and team member. They're issuing requirements and constraints, but they need to engage with the project more than just at that level. They should be (or at least feel) involved with the decision-making process.

They are team members because of their level of commitment to the success of the project. If they're truly working on behalf of the project and are interested in seeing it deliver what it's supposed to deliver and execute well, then having this kind of person on your side is a wonderful boon to your project. If, for other reasons, they don't want the project to go well, then you still need to treat them as member of the team.

I once watched a project that was intended to rewrite the way my company did sales reporting. As you can imagine, sales reporting is a very critical, pressure-laden application. The company's financial status and employee compensation were tied to it, so speed and accuracy were important for the system. The current system was pretty bad. It was down about 10% of the time, and although everyone agreed it was reasonably accurate, no one thought it was exactly right. Thus the project had the advantage of trying to replace a flawed system.

As it happened, I knew Dave, the guy who was the manager of the sales-reporting team. I didn't ask him about the project specifically, but I knew he'd be interested. So when I bumped into the project manager, I said, "Hey, how's it going with Dave? He's a pretty good guy to work with." The project manager looked at me a little blankly and then realized who I was talking about. She replied, "Uh, we haven't looped him in yet." This project never launched.

If the functional manager is on the team, then you're safe in treating them as a team member and probably using them as a reviewer. But if they aren't, you still need to respect their interests and treat them as you would another team member. This includes invites to meetings, regular communications, and making them feel like part of the decision-making process. If you fail to do this, you create a barrier to the project that you must overcome—a barrier that didn't need to exist in the first place.

Analyst

I don't believe I've ever uttered the following words: "This project has too many analysts assigned." Well, maybe I have, but I probably didn't believe it. A good analyst has a role on a project that is irreplaceable. A bad analyst can steer a project in a bad direction.

An analyst has two separate but related duties. They must gather and collate data in a way that can be used to make decisions, and they must formulate a point of view or a recommendation from the data collected.

The complication I've seen is that many people, on the project or not, feel they should have input into project decisions and prioritization. A good analyst knows how to best build a case using data and facts, provide some analysis, and allow the decision-maker to have the final say. But many projects and organizations are filled with amateur analysts—people who know how to mine data but don't know how to make an actionable recommendation from the information they collect.

Someone who can collect information but is unable to synthesize it enough to help drive decision-making isn't very helpful. Likewise, someone who feels free to make recommendations that aren't supported by data isn't performing a useful service. This issue can take two shapes. The analyst may draw conclusions that are actually in opposition to the data they've collected, or they may make recommendations in the absence of data, going off their personal knowledge and experience.

This is a very fine line for a project manager to walk. Analysts (and those who aspire to be analysts) are vital to a project's success. Without good data and good analysis, you may not have the information and support to make the decisions the project needs. That said, beware of the following two types of analyst: first, someone who is unable to accept that projects accept multiple inputs and that even though their conclusions and recommendations are accurate, they're missing a piece of information that they has no real way to include; and second, someone who wants to participate and make decisions based on something other than the output of their work. Both types can be dangerous to a project.

■ **Note** Be wary of people who are willing to provide conclusions that aren't based on data.

Here's my favorite example of the first kind of problem. I consulted for a company that was experiencing fraud: customers were ordering items, receiving them in the mail, and then scheming different ways to not pay for them. From a financial standpoint, this was a nontrivial issue. The company was losing six figures of sales every month, and the problem was accelerating as the fraudsters became bolder.

After looking at millions of transactions with several different attributes and drilling into each characteristic of the orders, our analyst said very definitively, "Overnight shipping orders placed on Thursday are the most fraudulent. If we stopped offering that service, we would save over \$100,000 a month." To be

fair, the data did support this claim. The theory was that the warehouse people were so incensed by on-time delivery that they knew orders placed on Thursday needed to be shipped, lest they be pushed into the weekend, and so they rushed to fulfill without looking hard enough at the orders.

Someone else in the room said, “Did you look at how many sales we’d lose if we stop offering overnight on Thursday?” The answer shined a very harsh spotlight on the analysis. The answer was, “Uh... not that much.” The unfortunate thing was that the conclusion had merit—a lot of our fraudulent orders were indeed coming as overnight orders on Thursdays. But the analyst was so firm in his convictions, without checking other data, that we eventually had to excuse him from the project. There was no way to move forward.

The second problem is much thornier. As a project manager, you need to make sure your team is balanced and that the roles are balanced. For instance, it’s okay to have a dreamer on the team, as long as there is the counterbalance of a Joe Friday—a “just the facts” kind of person—to bring balance. In most of my projects, I expect my analyst to play that role. If someone has a fanciful idea or wants to make a decision based on feelings, turn to the analyst to see if there is any way to substantiate the decision. If the analyst decides to put aside the rigor needed to make fact-based decisions and dream along with the other person, then you’re in trouble.

■ **Note** Balance in team construction is at least as important as the team’s actual makeup.

Here you’re dealing with a human-nature problem. All intelligent people like to engage in problems, even (maybe especially) problems that they aren’t equipped to help solve. I personally have next to no design skill, so when it comes time to unveil the creative output for the project, I’m willing to express my opinion but no more. I don’t claim to know whether or why something will work. I’m an expert in the statement “Looks fine to me,” but if I try to go into any more detail, I wind up making a fool of myself.

This doesn’t just occur for people who carry the title Business Analyst. People in every walk of life feel like they’re experts at something that seems obvious to the uninitiated, such as website design or product quality. Analysts are specifically trained not to allow their own personal biases to overrule the data they have collected or the conclusions the data can lead them to. Some people on your team will have the ability to do analysis—look at business problems and formulate a point of view and a recommendation—

but they aren't trained in the same skills. Often, these folks hold very strong convictions but without the proper discipline to ensure that their convictions are accurate, or at least based on the information collected. My experience has shown that these kinds of people stick to their ideas more strongly than those who have evidence to support them.

If you have analysts on your project, either by title or role, be sure they know that being an analyst means collecting data and formulating an opinion. Also, make it clear to them that they're only one input into decision-making. Analysis of facts can only take you so far; if you only followed what you know for certain, you'd never take the appropriate risks in the business. Finally, make sure they recognize and accept their role and don't drift into other, less data-driven positions. If you're counting on them to be the voice of reason, you must make sure they stay in that zone.

Developer

I'll use the word *developer* as shorthand for "anyone who is producing stuff on your project." If you're running a project that involves writing software, then this may literally be a developer, but it doesn't have to be. It can be anyone from a tax specialist to a writer. Anyone who can take a specification and deliver to it is a developer for your purposes.

Developers should make up the bulk of your project team. You need people who are able to take a specified task and go off and complete it. If these developers are team members, then they should also have a sense of why they're performing the task and where it fits into the overall delivery plan. If they're only contributors, then it may be helpful for them to know why they have been given the task, but they generally don't have time (nor do you) to try to understand the bigger picture.

Developers can also have opinions and thoughts about the wisdom or quality of a certain course of action, and they should be encouraged to express them. However, like analysts, they need to be made aware that they aren't the ones making the decisions; it's also fairly rare for them to have the full picture.

This often poses a different problem than the one you face with analysts. An analyst who thinks the project team is making a wrong decision can employ many strategies for making their feelings known. But in the end, they aren't responsible for carrying out the results of the decision. On the other hand, developers need to execute on decisions, and the decisions may very well be ones they disagree with. And truth be told, the developer may be correct.

But it's important not to let this become a self-fulfilling prophesy, where the developer doesn't believe in the task they're assigned and therefore doesn't carry it out to the best of their ability.

The strategy to help with this is complicated. You, as project manager, don't need the team to agree with every decision made throughout the entire project. But they must carry out the decisions made, even if they don't completely agree. You must achieve this without a team member gaining the mindset of a contributor, although in essence that's exactly what you want—but only for this task. The developer must continue to focus on the project and be ready to contribute to the next discussion, even though this one didn't go their way.

Tester/Reviewer

I'm listing these two roles together, even though they're often filled by separate people. The job of a tester/reviewer is to make sure that whatever the developers produced was done correctly: that it both delivers the requested functionality or use cases and matches the standards for acceptable work in the organization. These needs may seem at odds with each other—the developer must get as much work done as possible while delivering all the needed features and working to the level of quality the reviewer requires. It's another tightrope walk.

A project manager who views these folks as the enemy or as obstacles that need to be overcome is missing the point. Unless they're extremely outside of your project scope, such as compliance or audit, you need to make sure these team members fully understand the project's goals and objectives. More important, they need to understand what is important and what isn't. This greatly influences how they respond as the project goes along.

The worst position to take, in virtually all cases, is that all requirements are the same. If everything is a must-have, including technical requirements, then you can't prioritize and figure out tradeoffs. Similar to the way you talk with your clients, you should use the vocabulary of importance and priority with your testers and reviewers. Of course, everything must be tested; and, of course, you want all work to be reviewed. But some items need a little more scrutiny than others. It would be difficult to disagree with this principle.

This is how you should talk with testers and reviewers. You need to find out what they think is important and stress that they should give those things the most attention. You should also have a list of items that aren't quite as important and that should get less attention. This isn't cheating—it's the right

way to prioritize, although the reviewers may not be used to it. Don't be afraid to explain that they should spend the majority of their time on the things that matter to them most. That's what you do with your team, too.

■ **Note** Proper prioritization goes for review and test, as well.

You must also consider the review of things that aren't important. Some elements of your project may be tested or reviewed and found to be not perfectly developed to specification, and you won't act on them further. This is the sad outcome of an environment with limited resources and limited focus. If you aren't going to fix defects, either functional or technical, then why test in the first place? You must explain the answer to your testers and reviewers.

Be sure to define the items that the project or business deems important, and be open to the list of things the reviewers think are vital. But that will leave elements that no one feels are critical or in need of testing and validation. Resist the urge to allow reviewers to vigorously test these items "just in case." If they feel they must look at these low-priority items, be sure you set the right expectation. Even if defects are found, there may not be time or resources to make changes. If you don't set this expectation up front, you're probably in for an argument later.

Client Interface

If you watch tales of what business was like in the 1950s or 1960s (say, on *Mad Men*), you see that no one talked to the client without the account manager (known as an "accounts man") present. Those sensibilities have long since passed, and you're usually in a situation where anyone can have contact and free, open discussions with the client. However, there was a reason things used to be done that way, and it isn't entirely outdated.

Being able to work with a client is a skill like any other. Some people are blessed with it; others take years to learn it. And we've all run into people who don't have it. That's okay—there are plenty of skills that each of us doesn't have or could be much, much better at. But although people are willing to hear something like, "Susie is better than you at database optimization" or "Ralphie is a better graphic designer than you," virtually no one is comfortable hearing, "Pete is a better talker than you are."

Clients, internal partners, sponsors, or executives want two things in their collaboration with the project team: a few voices speaking, and speaking a consistent message. You don't want to outright prevent your team members from speaking with the client (in fact, you probably can't), but you should make sure the content of the conversation stays aligned, no matter who is speaking. And yes, you may want to prevent certain very underskilled team members from updating people outside the project.

The easiest and most effective way to do this is by being direct. Make it clear that anyone can talk with anyone else, but the content of those discussions should contain the message you want to send. In team meetings or other team communications, say things like, "If anyone happens to talk with the client, be sure to explain that..." or "If anyone is contacted by the project sponsor, feel free to send them my way for any updates or questions."

■ **Note** No one ever regretted getting a clear and direct explanation of what their role on a project was supposed to be.

Some people on your team will be fine. Many of your senior folks will have experience dealing with these kinds of conversations, and you shouldn't discourage them from continuing. But again, the way to do this is to be direct. The first time Sally asks your permission to meet with the client, let her know, "Sally, I trust you to do this without me, but let me know if anything weird comes up or you want my help afterward." Sally now has the green light to use her judgment, which you hope is good.

The client-contact role isn't for everyone, even though many team members probably feel they should have it. You can't hope to keep each person on the project team completely apart from everyone on the client side. But you can explain the importance of staying aligned in the conversations, and allow skilled team members the freedom to represent you and the project.

Policing Roles

Your team includes a mix of leaders, members, and contributors. It's critical that people know their roles so they know how to act and what is expected of them. It's equally important that you guide people back into their roles when they stray out of them.

I used to have a really sharp engineer on my team. He was universally accepted as one of the best in the company. Problem was, he was only 26 years old, and he didn't feel like a leader. But there was no doubt he was *the* leader on his teams and a leader in the entire company. Every so often, he forgot that his role was to lead and that other team members were looking to him for cues on how to behave. When he subconsciously slipped into team-member mode, the project suffered greatly. It took gentle pressure to remind him what his role was and how he needed to play it.

This kind of thing happens all the time. Team members act like reviewers, analysts decide to take their findings directly to the client, and leaders behave like contributors. This definitely adds to project drama and is something you need to avoid. It's a double negative: not only is someone acting in the wrong role and causing confusion, but also no one, or the wrong person, is performing some other role. Confusion and lack of role clarity are sure to cause your project stress.

At the start of the project, or at the beginning of a team-member engagement with the project, have a very clear discussion with them. Describe the role you expect them to play, and tell them that if you see them acting in the wrong role, you'll say something. It's important to set both of these expectations before the project begins. And you need to follow through: make certain everyone is doing what they're supposed to.

If, along the way, you discover you have leaders who can't lead, or client-facing members who shouldn't be in that role, or a team member who has skills you didn't know about, then you need to make a change. The worst thing you can do is have people playing out of position due to need or gaps in performance. Your project will go much more smoothly with people assigned to and performing the right roles.

Summary

In addition to carrying out the traditional project-management responsibilities, a good project manager knows that they need to structure their team in a way that sets up the team for success and limits drama and stress. This means constructing a team that aligns with the needs and objectives of the project, that can execute on the tasks that need to be completed, and that can communicate how they're doing in a way that the project manager and team can understand. Almost all the items you need to consider will help or hurt one of those three attributes.

In addition, consider the levels of the team members. Be sure you have enough people in leadership roles, a large body of team members, and the appropriate number of contributors to execute specific project tasks. Every project is different and needs different numbers of each type of person, but there is little doubt that your project will need people of each kind.

Understanding the role that each member is supposed to play, from reviewer to developer to client engagement to team leadership, is also key for a stress-free project. Setting each member's expectation about their role and how you plan to manage them in that role is key to staying sane. Keeping your eye on this is important for a smooth-running project. If you can't provide clarity to the project about who is doing what, you wind up with confusion and drama. And that's what you're trying to avoid.

Handling the Truly Unexpected

“Uh oh...”

Project managers are often viewed as professionals who create a plan and then execute to that plan while tracking status and providing updates. Some would say that the role of the project manager is to navigate the plan and the schedule while doing their best to avoid changes or difficulties. You probably know this is never the case. If you could simply create a plan and then execute it without any changes, adjustments, or problems, then your job could be done via an Excel macro. And I suspect you know that most of what you do every day could never be done with a few keystrokes.

■ **Note** Project management is a lot more than color-by-numbers.

To be fair, there is a role under the project-management umbrella for people who do exactly what I just described. The act of taking a project plan and giving an accurate summary of where you are is a nontrivial task. But this task of merely managing the plan and schedule can be done without knowing the goals or objectives of the project, and without once communicating with the client or the team. Depending on the size of the project, this role is vital to a smooth operation. It's not what I'm talking about here.

When I talk about project managers, I mean people who manage the project. It seems silly to say that, but there is no better way to describe the job without going back to the definition. Management requires active guidance and direction for all routine and non-routine tasks being done that further the aim's objectives. It could be a project is very routine and doesn't need a project manager, and an Excel macro will suffice. But real project managers never seem to get those projects, nor would we want them.

Project management is about the ability to handle non-routine problems and keep your project on track and creating the value it's supposed to create. As you've seen in past chapters, some of these non-routine problems really are routine. Challenges in communication or in client alignment plague more projects than not. Difficulty in managing scope or creating success criteria isn't uncommon either. But sometimes new and unique problems occur within a project, and it's up to you to figure out what to do about them. This is when you earn your living.

There is also a class of problem that you can't foresee but, with hindsight, should at least have been somewhat prepared to handle. This is a difficult balance. On the one hand, you couldn't have known the problem would occur, either due to lack of experience or because you're attempting something truly new in the marketplace. On the other, with the knowledge you now have, you realize the problem was a direct outcome of something you did or failed to do. This is a great thing to have happen to you, as the only way to learn the unknown is through experience.

One of the reasons you manage projects is the ability and opportunity to do and learn new things. You can move from industry to industry, from one problem domain to another, and carry your skills with you. This allows you to gain more domain expertise while sharpening your core abilities to get good work done. One way in which you grow as a manager is by facing and overcoming problems you've never had to deal with before. And you can only do that if you're open to the idea that problems will occur and that it's your responsibility to figure out the resolution.

Not only do you gain experience and knowledge that enable you to overcome these problems, but you're also prepared for them in the future. You can help train your peers and team members about how to recognize and be ready for these kinds of interesting problems, making you more valuable in your project-management community. Having interesting problems makes for a good story; having routine ones doesn't.

No-Drama Preparation

The continuing themes of no-drama project management include communication, intentional decision making, and determining success criteria before attempting to create a solution. Another theme has been to expect problems that should have been expected. In this case, it means preparing for the problems you think may happen, before they actually do happen. It's surprising how much value you gain by preparing for issues to arise, even in ways that may not be apparent.

A friend of mine ran a project that involved getting a huge amount of raw materials—literally tons of the stuff—for a manufacturing facility in Europe. The vendor-selection process had resulted in choosing a supplier in Hong Kong. The project manager, my friend, was certain this wasn't going to work. Although the vendor was fairly well established and seemed to be saying all the right things, he doubted they could actually supply all the material he needed. He even told them that straight out. They told him not worry.

But being a good project manager, he did worry. He crafted a few scenarios around the vendor's inability to deliver, such as a partial delivery, no delivery, or a very late delivery. He also suggested that the vendor might ask for more money from us, in order to express-ship or otherwise expedite the material to the plant, and he worked out what that would do from a cost of goods sold (COGS) perspective and how it would impact the business case. If the vendor was going to be late in producing, he was ready.

As it turned out, the vendor was true to their word. Right on schedule, a huge container full of materials was ready to ship. It was travelling by tramp steamer (or something equally slow) to a European port, but it would arrive in plenty of time for the market launch. Assuming the ship wasn't attacked by pirates,¹ there was nothing to worry about. My friend crossed "vendor delay" off the list.

A couple of weeks later, the container arrived at the port. This was way ahead of schedule, furthering the delight of the project manager. He wasn't there personally, but he sent someone from the plant to get it. Later that day, the guy who went to the port let the project manager know that the container hadn't cleared customs yet, and he'd have to go back later. No big deal; it happens all the time.

¹ A great example of an unexpected problem.

But weeks passed, and the container was still stuck in customs. No amount of calling or pleading changed things. There was almost certainly nothing wrong or unusual about the container, but it wasn't cleared. Of all the things my friend was tracking, "failure to clear customs" wasn't one of them. We had done this kind of shipment several times before, and we expected to do so once a month for the foreseeable future, based on our understanding of demand and how often we would need to replenish the inventory.

This could have been a disaster. The project manager could have thrown up his hands and said this wasn't anything he could control, and that customs delays and other governmental-type slowdowns weren't the project's fault. Because everyone agreed that he had done everything correctly and the whims of a few border officials were causing the problem, he would have been mostly in the right had he done so.

But that's not what the project manager did. He recognized that the actual problem wasn't a delay at the border: it was the fact that the plant didn't receive the raw materials in time to begin production. Although he wasn't prepared to deal with customs officials, he was prepared to deal with this kind of delay. In fact, he had expected it to occur—only he thought it would be the fault of the supplier in Hong Kong. This is an example of not being prepared in specific, but being prepared in general.

■ **Note** Sometimes it's worth it to be prepared for outcomes, not causes.

By being ready for the situation, the project manager was able to pull out the "late to plant" planning he had crossed off months before, and move on it. Rather than cause an excess of anxiety, he was able to say, "We weren't prepared for *this*, but we were ready for something just like it." That immediately diffused the situation and moved the project away from blame and stress and into executing on a solution plan that was ready to go.

Was Your Problem Unexpected?

This leads to the next question your program manager or client will ask you: "How unexpected was this problem?" Although it can be difficult to prepare for the myriad causes and oddball occurrences, you should be prepared for the impacts on the project. In the case just described, the impact was the late arrival of the material, not the customs delay. I highly doubt a

good project manager would get away with making the assumption that the supply chain would work flawlessly the very first time they tried it.

This kind of thinking should permeate your planning. Chapter 4 discussed the problems you may face when determining project priorities. There are many reasons this could be the case—many more than I recounted in that chapter, and probably more than I could fit into this book. However, it should be no secret to any project or program manager that difficulty in prioritization should be expected—or at least shouldn't be unexpected. You may not know the cause, but the cause isn't nearly as relevant as you think.

Of course, it's possible to have problems that were either unexpected or completely unknowable when the project began. Such issues make the job interesting and teach you new things. Sometimes they're particular fine points that are linchpins holding everything up. Other times, they're mammoth problems that no one was aware of until you got started. Or they may have next to nothing to do with your project.

A colleague of mine was running what seemed to be a fairly benign, nearly paint-by-numbers project to add more stock art to our artists portfolio. We had done this many times before, and the process was neither hard nor fascinating. He assigned himself this project because he knew it wouldn't be difficult, and he could run it along with all the other duties he needed to manage. As it turned out, he was wrong about how much time it would take, but he was glad he had called his own number, because the problem was pretty thorny.

■ **Note** Truly unexpected problems take senior management and firm guidance.

It turned out we had a huge database of stock images that we had bought a very long time ago from a company that was at the time fairly new. We had signed a very aggressive, very favorable license that the supplier would never be willing to sign again. As part of adding the new library to the package, we would be slightly altering how we were using this supplier's images; this would represent a change to our license.

This company was always looking for a reason to change our license, but none of us knew this. It never occurred to us to put on the list of project assumptions something like, "Vendor may refuse to sell to us even at full list price." Even now, I don't think we'd list that—it seems like the kind of thing

that's fair to assume in the normal course of doing business. Well, in this case, the supplier was agitated enough that they wouldn't sell to us at any price.

This was causing a bit of a problem, because we had begun making commitments to clients for products that were made with this art. The vendor probably didn't know this, so they weren't trying to stick it to us—they just wanted out of a very unfavorable contract. We had no backup plan. We were stuck. Once we realize this, we had little choice but to raise the issue with our client and sponsor.

We didn't own the relationship with the vendor, nor did we own the relationship with the end consumers who were buying the products made from this package. All we could do was seek guidance for how and whether we should proceed. It's one of the few times in a software project when I had no advice to give. No amount of schedule or budget changes could solve this problem. If the vendor wouldn't work with us, and the vendor manager on our side wouldn't budge, then we should go do something else.

Before I get into the strategies for what to do next, it's important to note how much drama a truly unexpected problem can bring into a project. There is a sense of, "What are we doing to do?" in the team, because every single solution is either out of their ability to control or possibly out of their scope to influence. The right thing to do at this point is not to say, "I don't know what we're going to do." The right thing is to explain that when a truly unexpected, material problem occurs in a project, there are strategies to follow, and you expect to follow one of them. Keep the team from worrying, because no amount of worry will make things better. Let them know you're aware of the problem and presenting strategies to the client shortly, and they should sit tight until that happens. Be sure to communicate with the team as often as possible; when no information is available, people tend to invent their own bad news.

Strategies for the Unexpected

Once you've determined that you do have a material, unexpected problem in your project, the right thing to do is to call it out to the sponsor, the client, or whoever is empowered to make a go-forward decision. The material problem may be an unexpected obstacle of some kind, a change in market conditions, or a drastic change in the environment. Whatever it is, you can't ignore it, and continuing down the path you're currently on won't lead you anywhere good.

■ **Note** Material problems should cause material changes; if they don't, you should be careful.

When this happens, your project has three avenues of approach: restart, revise, or rescope. Each is fraught with its own challenges, and each has its own strengths. There are many factors to consider when determining which is the right way to head. Let's look at each one.

Restart

Restarting a project is the most drastic of actions and therefore the one I discuss first. This is usually caused by a shift in the business case for a project. This may mean the project's costs have increased or its expected profitability has decreased. Either will greatly impact the project's return on investment (ROI). It may also be that the environment or client priorities have changed, causing the project charter's initial foundation to no longer be valid. Whatever is the reason, the original value of the project is no longer true.

From a project-management perspective, you're running a new project. Granted, the project deliverables may look the same, and the plans you were going to use are probably still close to accurate, but the project's value definition is different, and the ROI calculation isn't the same. This should set off warning bells in your head, and you should take care to revalidate that the project is worth continuing. If you decide to continue it, you should gain an understanding of where the project falls in the priority list. If it used to be the top project in the portfolio, it was probably resourced and supported in a specific way. If it's now in the middle or toward the bottom, you can probably expect different treatment. You need to know before you're surprised.

A company I once worked for had a project to start construction of novelty products. We believed we could use our existing manufacturing lines to begin making a new kind of product: one that would hit a different market than we traditionally sold into. This would be an interesting way to take advantage of existing capabilities. Without having to change anything, we could create products that would sell to a new customer base, thus ensuring that we weren't cannibalizing our existing sales.

It then turned out that our existing manufacturing lines couldn't make the products we had hoped. Something about tolerances or clearance—I never quite understood. Anyway, new equipment would need to be sourced and purchased, floor space found, and operators trained and hired. The expected

revenue for the case remained the same, but the cost structure took a drastic turn for the worse.

From the business side, we probably did the right thing: we reforecast the COGS, lowered the profit expectations, and published this new information far and wide. We were open and honest—perhaps too much so. What we did *not* do, however, was change the attitude of the project team members, and this caused significant drama later in the project.

When the original project began, people wanted to be assigned to it. It was a pretty big deal—a new product line, great profitability numbers, and a level of team (both in numbers and qualifications) that matched the importance of the project. The team wasn't overly arrogant; but members knew they were a good team on a good project, and they acted as such. As we moved forward with the new project, which had lower profitability and priority, the attitude of the team was in conflict with the project charter.

As one of the lower-value projects in the portfolio, it was sacrificed constantly to make more important projects better: one or two of the best people were reassigned to other projects, requests to other groups were serviced more slowly, and so on. This frustrated the team, which was still acting as if it were working on the first project instead of the second one.

By “second one,” I mean we should have restarted the project, and perhaps given it a different name and a different team. Other than the final product, very little stayed the same from the initial project to the subsequent one. The reason we didn't restart the project or completely recast it as something else was almost certainly a failure in client management. And that's the key point.

■ **Note** Restarting a project is more of a client-management challenge than a project-management challenge.

Restarting a project is difficult. Often, it's more difficult than starting a new one. A new project has the advantage of nothing having gone wrong and no disappointments to report, no matter how small. Taking a project that was green-lit to run, and going all the way back to business-case creation and re-approval, puts the entire project at risk. The client or sponsor is understandably reluctant to do this. Any time you bring a project back for approval, there's a chance it won't be approved in its new form. No client wants to hear that. But as a project manager, you must get someone to make this decision.

If you don't take the project back to be reapproved in its new state—or if you're legitimately worried that it won't be approved if the truth is known and examined—then you're essentially running a project that was never approved. It's hard to imagine something that can cause more confusion and chatter. If the project ROI calculation no longer works, you shouldn't be running the project, period. Hiding this fact does no one any good, and it will probably cause harm down the road.

On reflection, if you think the project needs material revision in value, scope, or cost, or is starting to feel like a different project to you, then you should work with the client to reshape it as a new project. You may lose this one in favor of something else. But if that's the case, better now than in six or more months, when either you deliver something that doesn't create the expected value or your team begins wondering why the project isn't as important as it once was. Remember, you want the client and the team to work with you again. Make the correct call at this juncture, rather than delivering something that no one is excited about.

Revise

Not every project that hits a problem or big change needs to be halted or restarted. If that were the case, you'd probably never get a chance to launch anything, because we live in a dynamic and changing environment. Sometimes it's possible to change a project in a way that doesn't have a meaningful impact on the project's value positioning. In fact, this happens more often than not.

This tends to happen with a project that needs to increase its scope by 5% to 10%, change the specific resources assigned, alter its schedule slightly, or suffer a one-time cost. Some of these things can also cause a restart: for example, if the one-time cost means that instead of being profitable in year one, the project now has a payback period of four years, that's a material change. Similarly, if you're planning to launch a product for the Christmas season, but a delay will push the launch into February, you have a serious change on your hands.

But in many cases, a simple revision is possible. Perhaps you need to delay the project by weeks, not months. Or the missing item you need to buy has a non-meaningful impact on the profitability of the end result. Except in the smallest of projects, revisions are more likely to occur than not. Most of what you're doing has never (or rarely) been done before, so you're constantly learning new things along the way: things that invalidate plans you

made months or quarters ago. This isn't a sign of failure or poor execution; it's a fact of the project manager's life.

This issue is one of the reasons companies deploy professional project managers for important efforts. In many organizations, project managers are well within their rights to shuffle the people on their project, change the project plan, or do whatever is necessary within the initial budget to deliver the project. This often takes a serious amount of creativity from the project manager and trust from the client and team. Usually, the amount of time spent analyzing a revision is a mere fraction of the time spent creating the initial plan, but you desire the client and team to have the same level of confidence in the revised plan as the initial one.

Nearly all projects go through some revision. Some project-management methodologies and organizations require that each change be removed from the original project and managed separately using a change order² or something similar. Sometimes it's possible to use the existing people in a different way in order to keep the project going.

For instance, once, in the middle of a project, we collectively decided that we should do some focus-group usability testing. The cost of the testing was nominal (the room and facilitator were fairly cheap), but it required that we have a mostly working prototype of the product for the subjects to do their testing on. Not only did this prototype not exist, but it was never in the plan to make one. It was likely to require two or three weeks of work if we wanted to get any meaningful results from the test.

The first step the project manager took was to understand the value of the request. Sometimes revisions seem very important and highly valuable because they're the latest thoughts someone had. Before you proceed, make sure you aren't running into the fallacy of "most recent wins." In this case, it was clear early on that everyone was on board with doing this kind of testing. It was a new product in a new market, and all we had was our own experience and best guesses to see if we got it right. Some of the people on our team were pretty sure our guesses were correct, but no one was willing to say they knew for certain. So, usability testing was scoped into the project.

■ **Note** It's important to confirm that revisions have the right level of value. It's easy to fall in the trap of "most recent request wins," even though that may not be the right thing to do.

² http://en.wikipedia.org/wiki/Change_order.

The second step was to figure how to pay for it, or in this case, how to re-source the building of the prototype. This would require weeks of work, and we knew it should be done by someone familiar with the project. If we brought in an outsider, the overall cost of the project would increase and we'd have to train the person and explaining the project's goals and objectives. Otherwise the prototype wouldn't be useful.

The client was unwilling to shoulder the cost of adding four to five weeks of an engineer to get this done, but they did want the prototype built. So, the project manager proceeded to the third step: determining how to use the existing team and budget to do the work. As it turned out, the project plan included some important but not urgent items, as well as a couple of not-important, not-urgent items. For people who know Steven Covey's prioritization categories,³ these are Quadrant II and Quadrant IV activities.

Eliminating a few of these tasks from the bottom freed up one of our web engineers and a database programmer. They weren't the ideal team to build the prototype, but they were pretty good, and they had the right qualities. They built a very useful prototype. The usability testing confirmed that our initial guesses were mostly correct, and now we knew for sure.

But this created a change in the original project. Some of the items we planned to deliver were off the project and might never be delivered. The project manager needed to ensure that the client knew this and wouldn't be surprised or upset when the project closed. If you have a very good, ongoing relationship with a client, you may get away with telling them something like, "Remember, because of the prototype, you're not getting that automated testing thing you originally requested." But even that rarely works, because one or both of you eventually forget.

This is where the revision process comes in, and it's a judgment call. You need to answer this question: "Is this a material change to the project?" This doesn't mean a normal change or alteration that comes through requirements refinement or deep-level analysis that wasn't available originally. Changing a user interface (UI) element from a link to button doesn't apply. It's also possible that removing bits of originally requested functionality doesn't apply, if it's decided that the functionality either isn't needed or isn't that important. Material changes are things that, if not delivered, will cause the client to ask for a refund.

³ [http://en.wikipedia.org/wiki/First_Things_First_\(book\)](http://en.wikipedia.org/wiki/First_Things_First_(book)).

■ **Note** Beware of being casual about important things.

If you think you're facing a true change in the project, you need a revision process. I don't necessarily mean a legal document that requires sign-off, or something that gets encased under glass for all the world to see—although, in some cases, this isn't a bad idea. You need to signal to the client that this change is *different* from others, and you want everyone to remember it later in the project—including yourself. A *project revision* is different than a small change, and it deserves a different kind of treatment.

Rescope

Rescoping a project falls between restarting and revising. With a restart, you're saying, "This is an entirely new project." With a revision, you're sending the message that the project is essentially the same, but with some important differences from the baseline. With a rescope, you're adding to a project, but without touching the original project. This is how most consulting companies stay in business: they sign you up for one project and then continue to add to it until the size of the additions is bigger than the original engagement.

That's not necessarily a bad thing, but it's rarely correct for it to be the default solution. Adding to a project almost certainly increases cost, and it's unlikely to add commensurate value. The additions may add value, but they usually don't do so at the same rate the project was originally intended to do.

But that is sometimes okay. Even though a change may lower the ROI of a project, the ROI may still be good enough. One of the questions I tend to ask my project managers when they want to rescope a project is, "If we had known this before we started, would we have approved the project?" The answer is in the affirmative a surprising amount of the time, which is probably a sign that many projects are intended to have a good enough return that they can withstand some added cost. My project managers know not to start their answer with, "No, but..." because whatever follows isn't relevant. Sure, we may decide to add scope to a project for important reasons other than ROI, but that's not what I'm asking about.

Similar to revising, rescoping should be done intentionally and transparently. Some project managers call whatever is in the newly scoped project "Phase 2." I often encourage them to name it specifically after the additional functionality. If you were planning to sell square widgets, and you've

decided to add the ability to make and sell round ones, then don't call the entire project Widgets. The additional work is either Widgets Phase 2 or, even better, Round Widgets. That makes the extension's purpose clear to everyone involved.

This is an important because sometimes the client or sponsor changes their mind about Phase 2. Cooler heads prevail, or they get a better idea, or market conditions change, and they no longer want round widgets (although they still wants square ones). Oh, and they're sorry for putting you through the trouble.

This is a good reason to keep the extension separate. If you decide to add a material amount of new scope into a running project, you'll wind up conflating the two. Interleave their planning, execution, and release, and soon enough, you won't have a project and a rescope: you'll have one project. Extracting the parts that formed Phase 2 will be more work than you anticipate.

■ **Note** Once two plans are merged, they're hard to unmerge.

Additionally, you want to know how much time has already been spent on the rescope part of the project—not so you can measure the sunk costs, but because you and the client deserve the visibility into what that decision and change cost. Even if you decide to run the two things together, you should keep your documentation, planning, and other artifacts separate, so you can tell one from the other.

Managing Fallout

As the project manager, you're likely to be asked something like, "How did this happen?" There is no way to avoid this question, nor should you want to. And there are as many answers as there are projects. In your mind, separate these two topics: "How did we get here?" and "What are we going to do now?" How you manage the former impacts your relationship with the sponsor or client, whereas how you manage the latter strongly affects your relationship with your team and management. It's possible to maintain your relationship with both these groups—and even improve it.

You must realize that these two questions have very different areas of focus. You may have been involved in the process that caused a big miss in the

requirements or solution design. Or perhaps an initial assumption turned out to be so wrong that it invalidated the entire foundation of the project. Of course, it's also possible that the error happened before you were involved in the project, or maybe the miss wasn't a mistake at all—it was unknowable until the project began. The key thing is that it doesn't matter. All mistakes are learning opportunities, and whether it was you who made the mistake is immaterial. How you got where you are is probably factually determinable.

For instance, I once consulted for a company that was in the process of buying another, much smaller, company in a different country. Originally, this was supposed to be a “buy and hold” acquisition. The purchasing company was going to buy the smaller company, merge the financials and balance sheets, and let the acquired company run as an arms-length subsidiary for the foreseeable future. This was probably a good idea at the time.

However, it became known in the industry that the subsidiary was owned by the parent (the company-issued press release didn't help keep this secret) and that the subsidiary had a much different level of quality than the parent. Mid-project, the sponsor decided that allowing the subsidiary to produce substandard products would do too much damage to the parent's brand. Therefore, the project was rescoped to include increasing product quality. This acquisition-quality project turned out to be bigger than the initial project.

It became relatively well accepted that this project was the right thing to do. It was also widely wondered how we ever thought we could *not* do it. But truthfully, at the time, this was a reasonable decision. The decision was made before anyone knew how public the announcement would be or how substandard the subsidiary's products were. It was an understandable error, but it was an error nonetheless.

Neither you nor the client should try to hide from the cause of a problem, but you also shouldn't let it affect what you do in response. This gets into the second issue: what to do next. Even if the problem was a complete miss and was specifically and solely caused by you, that doesn't mean you shouldn't create the plan; you're still the project manager, after all. Don't wait for permission or seek approval to create the plan for what to do next. It's your responsibility until you're told otherwise.

Being able to segregate the cause and the solution is a vital skill not just for this occurrence but also for many things in your projects. Often the cause isn't known, but you're still on the hook for crafting a solution. Likewise, the cause can be so far removed from the actual impact that casual analysis

doesn't yield anything useful or actionable. Be clear about not getting hung up on who or what caused the problem, and instead focus on how to work your way out of it.

Managing the Team

How you react to problems that are truly unexpected is an important factor in the success or failure of a project. If you let the decision take too long, or if you continue to plow ahead as if nothing is amiss, you'll begin to lose your team. They will very shortly know that something is wrong, and pretending not to notice won't calm them down. In fact, it's likely to cause them to think you don't know what's going on, which will bring drama to your project.

Once you recognize that the project you're running can't be successful in its current state, you need to do something else—and you need to do it in a constructive, open way. Continuing on a doomed path will win you no favors with your client or your team. And attempting to “fix” it in secret is also unlikely to curry favor with anyone. Of course, you don't want to shout “My project is failing!” over the corporate intercom, but your team and client know that changes are coming, and you should be up front about them.

This is even more important between the time that the need for change is obvious, and the change is actually made. It's not uncommon for this span of time to last for weeks, and those are precious weeks for the project. If you create a situation where your team is dispirited or doubts the project will continue, or if you let them think there is no reason to continue working, then you've harmed your project in an unforgiveable way.

I acknowledge that this is a tightrope walk. But it emphasizes why you need to be open and honest, as well as clear and decisive. You still have a plan, and most of that plan probably remains valid. If you can keep your team working on the things you know will continue and add value, then you should do so. If you can't, then maybe the team shouldn't be working on the project. A few days of busy-work and writing project documentation won't hurt anyone, and the breather may be welcome. But if days stretch into weeks, or weeks into months, then you've harmed your ability to get the team moving in the proper direction again.

■ **Note** Remember the needs of the team while decisions are being made.

For this reason, rescope or restarted projects often lead to a *go-forward team*. It may not be a good idea to try to pick up and continue with the exact same team with the exact same makeup. The team may be missing a needed skill, or the new schedule may require different availability. But it's also a wise idea to bring in one or more people who don't know the project hit a bump in the road and is now on its second chance. You still have team members who are more motivated than ever to make the project a success, mixed with new blood who are energized by the project.

Summary

Despite the theme of this book—that many of the problems projects face were predictable and probably avoidable—many issues fall into neither category. If all problems were predictable and avoidable, the project manager's job would be pretty boring and unimportant. How you recognize and deal with these kinds of issues is what separates a project manager from a project administrator or secretary.

You can help in these situations by being prepared. Although it sounds like I'm saying you need to be prepared for all unexpected possibilities, that's not quite the truth. In many cases, you can't predict the cause of a problem, but you can predict the impact of the problem ahead of time. Consider the example in which my friend knew it would be a problem if materials arrived late to the manufacturing plant. Even though the reason the materials were late was totally unexpected, the notion that it might happen led the project manager to be prepared with a plan for what to do in that eventuality. Having that plan saved a lot of stress and drama from creeping into the project.

Before you add excitement and worry to your project team or client, make sure the problem you're facing was truly unexpected, and not just a flavor of a problem you're prepared to handle. If you decide to treat this issue as something no one could be ready to deal with, then you either needlessly raise your team's anxiety level or look foolish in front of the team when you discover that the solution was thought of before the project began.

If you face a problem that you can't solve with a predetermined strategy, and it's material enough that it changes the shape of the project, you can pursue three different avenues: you can restart the project, you can revise the project, or you can rescope it. Each strategy has its place in the project manager's arsenal.

If the new information about the project causes such a change that it's really a different project, then restarting it may make sense. And if going back to

the beginning means the project doesn't get approved, it's better to determine that now rather than after you deliver it.

If the problem to be overcome is solvable within the current project budget, as long as a few things are removed, then consider revising the project. This works best when the problem to be solved is of undisputable high priority and the project already includes some lower-priority items.

If neither of these strategies will work, then you're probably facing adding time and effort to your project by rescoping it. Rescoping isn't always a bad thing, but it's a sign of something big enough to need its own name.

At some point in the process, the project manager must deal with the fallout, even if they weren't involved when the mistaken decision was made. Mistakes are chances to learn and improve and shouldn't be used as ways to blame and point fingers. Each mistake makes you smarter for the next time, so a thoughtful discussion about what decisions were made and how mistakes could happen can only help you and your team in the future.

This analysis of the problem shouldn't impede your determination of the solution. They're separate exercises. Conflating the two will more than likely yield a disappointing result and cause more problems down the road.

Creating solutions for your team structure and the problem itself are both important to a healthy project going forward. You should call out the problem in the first place to help steer the project back on the path to success—an act that can be disruptive. Using this intentional disruption to set things right is a good use of time and energy. Doing it intentionally and openly helps keep your team engaged and avoids drama during the process.

The End of Drama

“Let’s do it all again!”

Chapter 1 talked about why you’re a project manager. One of the most compelling reasons is that your job requires you to do new things, learn new topics, and, often, launch new products. It’s a job where the principles remain the same, but the problem domain, industry, and challenges are ever changing. Being able to take your old skills and apply them toward new and different frontiers is something very few other positions can claim.

But that comes with a cost: you must always be improving your skills as a project manager and your general and industry expertise. Along the way, you’re allowed—indeed, expected—to make lots of mistakes. However, you’re also required to not continually make the same mistakes and, ideally, to never make the same mistakes twice. If you work in a program or a multi-project manager environment, this can be even more challenging—you’re expected to not make any of the mistakes any other PM has made, either.

■ **Note** Project managers are expected to make mistakes. But only once.

To make this work, you need an environment in which you’re learning from each other and viewing each other as members of the same team. While you’re all dealing with different clients and working on different and existing projects, you’re tied together by the thin thread of being accountable to each other. Don’t discount the value of having someone to talk with who has been in a similar situation. Sometimes you’re doing the talking, and

sometimes you're doing the asking. But either way, the more you can make your team into a team of project managers, the better off you'll all be.

All project managers aspire to have projects run with less stress. This doesn't mean every project is a rousing success or a shining example of a perfectly executed plan. Rather, it means you faced problems, just like any other project; you handled them in a way that didn't cause more anxiety than necessary; and you didn't get in the way when trying to navigate a difficult situation. In short, you're looking to run projects without the drama that sometimes comes along for the ride.

No-Drama Project Management

Throughout this book, I've talked about the many ways a project manager can contribute to the drama of a project. What I didn't discuss specifically is how to avoid drama entirely. There is no recipe to follow or checklist to keep in hand that can help you avoid the kind of unproductive anxiety that can arise when things are going wrong. But I hope you picked up on the key themes that will limit the drama among your clients and team members. They include preparation, being open and honest, and treating communication as a project deliverable that needs to be managed like any other feature.

Preparation

One of the quickest ways to create drama in a project is insufficient preparation. As Chapters 10 and 11 discussed, this covers both traditional project planning and what I call *no-drama preparation*. The former includes normal project-management artifacts such as dependency plans, work-breakdown structures, Gantt charts, and other elements that are basic to project management. The latter is a bit more nuanced.

Being prepared doesn't mean you have every possibility covered, or ready-made plans for any eventuality. There is no way to predict every single thing that can go awry in a project; if you try, you'll never have time to manage the project. However, it's significantly less challenging to prepare for the *outcomes* you want to avoid.

For instance, if you have a key person on your project, you may be concerned about what would happen if they weren't around. This could happen in a variety of different ways. The person could quit in a huff, win the lottery, or be injured in the corporate softball league. They could be reassigned to a more important project or be called on to rescue something else. Maybe a

high-ranking executive wants to recruit the person onto their team. The number of possibilities is too high to count—and this is only one risk.

So, don't worry about what can go wrong. You'll drive yourself crazy if you try. Worry about how a problem will impact the project, and what you can do about it when it happens. The good news is, this is a lot easier to manage and provides the greatest flexibility. The bad news is, it means your preparations require some creativity to execute. Because your plans will be close to the problem at hand, instead of the exact problem at hand, it will be up to you to make the mitigation strategies fit. But that's why project managers are assigned to projects in the first place.

Being Open and Honest

These are two separate things, because you can be one without being the other. Being open is about being transparent with your team, your client, and your organization. If things are going the way you hoped, or maybe they aren't, your team should know about it. One of the worst feelings as a team member is the sense that things are happening and everyone knows about it but you. Even if it isn't true (perhaps especially if it isn't true), it creates a sense of paranoia in your team member that is never positive.

There is value in letting team members see you manage. Sometimes your instinct is to "take something offline," but you should instead do the management where the team can see it. This can also mean letting them sit in on planning sessions, or sharing with them what's on your mind. There's a limit, of course, but you should bring your team into the process a lot more than you probably are currently. The more open your team feels you are with them, the more open they will be in return.

Being honest is a totally different challenge. My experience tells me that everyone appreciates the truth, even when the truth is bad news. Few people like being tricked or deceived; it doesn't matter if your intentions were good, such as to protect a team member or delay an unpopular decision until more information can be gathered. Truth that is negative isn't any less true, and your team understands this.

■ **Note** People mind criticism a lot less when it's true.

The project managers in my program send out status updates. Some do it weekly, some do it monthly, but they all do it. One of them added a stop

light (literally clip art of a stop light) to show whether a project is red, green, or yellow. I didn't ask for this, because I felt I could read the update and decide for myself whether the project was in good shape. One week the project manager sent out an update with a bright green light at the top, but the details in the message led me to believe the project wasn't green. I didn't stress over it—I had the details I needed—but I had to ask, “What is keeping you from calling this project yellow?”

The answer was incredibly interesting to me. The project was indeed in a little trouble (it was temporary—the project was back on track within three weeks), and almost everyone knew it. The team had a meeting to talk about the risks they were facing, and they came up with a pretty solid plan for dealing with the problem. They were acting on the plan, but they were a bit less than half done when it was time to send out a new update.

The project manager told me he didn't want to dispirit the team by sending out anything other than green. Sure, the project was a bit rocky, but they had come up with a great plan, and the team was working hard on it. Why broadcast to the greater community that the project wasn't going well, when everything either was under control or shortly would be under control?

I understood and appreciated this sentiment, but I thought it was wrong and that his client and team would see through it. The client was involved in the planning process along with the entire team, and they all knew something was going on. Deciding to call the project green anyway was disingenuous and not entirely truthful. One of the team members told me, “Maybe he's grading on a curve.”

The funny thing is that doing this, although well-intentioned, caused the project manager to miss out on several opportunities. The first was the chance to spark the team when the project slipped from a legitimate green to a solid yellow. There is power in telling your team you're in a mess and asking that they pull together to get out of it. If you tell the world everything is fine, you lose the ability to play this card. Even more depressing was the opportunity lost when the team pulled out of it, which I knew that they would. Once they set things back on course by working hard and tracking to their plan, no one knew it. The team missed a chance to be congratulated, because no one outside the team knew they had performed an impressive project rescue.

■ **Note** Your team will support you if you say you need support. Otherwise, they will assume nothing is wrong.

The good thing about the truth is that you can show it's true, and no one can get on your case for telling it. Better to be up front and truthful about project status, team mistakes, or anything else that is going on. You can always provide context along with it, but it's very difficult to explain a statement that turns out not to be true.

Treating Communication Like a Deliverable

This last point is both the easiest and hardest to grasp. As discussed in previous chapters, for most of the project, communication is all the client receives from you. This is the way your client, sponsor, and team judge how well you're managing the project and how on top of everything you are. Therefore, treat your communications like any other project deliverable.

This means gathering requirements ahead of time, understanding what is important, and following up after the fact to measure value. If you're sending out a communication to more than one person, the chances are incredibly high that your audiences have different needs. So, you're unlikely to meet all their needs with a one-size-fits-all strategy. If your audience measures in the dozens of people, you may need to say the same thing multiple times in different ways and venues. This is a good thing.

Communication in the project sense involves making someone else understand what's going on, what the key decisions are, and where the project currently stands. Because everyone has a different need and a different area of focus, you should take the time to understand how each person wants to be communicated with, how often, and by whom. Along the way, continue to ask if your messages are being understood and reaching the correct people. If you leave your communications to chance, it will only be luck if you happen to get it right.

Fortunately, you know how to create deliverables. You can figure out what you need to do, specify it, deliver it, and then measure it after the fact. This is what you do most of the day. Once you acknowledge that project communications are vital to the project and deserve to be treated as a first-order requirement, you can shift your thinking about them. Rather than being a task you need to do that takes up your time, they become something you get to deliver and on which you're judged and measured. The shift is palpable, and it creates a much different quality of project. It's a change well worth making.

The no-drama project manager spends extra time on being prepared, being open and honest, and treating project communications seriously and with

the respect that they deserve. These are great ways to insulate your projects from being full of drama. There is no magic bullet to prevent drama ahead of time, but doing these things can help from the start.

Making Intentional Decisions

Another theme of the no-drama approach is the notion that decisions made intentionally are usually the correct ones. Conversely, although decisions that are allowed to happen or are made passively may be correct, that happens only by accident or coincidence. Often, decisions that are left to linger wind up with a limited set of solutions when decision time is at hand.

Don't let things happen by accident or allow entropy and inertia to take control and decide for you. This allows decisions to go on so long that the outcome is finally forced on the project, or the developer makes a choice without thinking about it, or only one possibility remains. Letting decisions make themselves is a pretty sure way to reach the wrong ends.

But more than that, making decisions intentionally aligns with the top-down approach to prioritization and implementation. Determining what is important and spending the majority of your time on those things most often leads to a good result. It also allows for the proper amount of time and diligence when making these calls—something you won't have if you wait until the last minute to decide.

Deciding things early and on purpose offers three important benefits. First, all options are available. Second, you have time to communicate and validate the impact of the decisions you're making. And finally, you can hold yourself and others accountable for the decisions, rather than hiding behind the façade of not having enough time or being in a rush.

One of the saddest discussions I have with project managers is when they know what the right decision would have been, but it's too late to go forward with it. A problem that had several good solutions now has only one remaining, due to time constraints or budget—or, worse, because a different but less important decision was made that now prevents it. The old phrase “keeping your options open” only works if you're actively keeping them open. If you're delaying decisions merely to delay them, then you'll probably be disappointed in the end.

As I've said in many places in this book, communicating what you're doing clearly and correctly is vital to no-drama project management. However, it's also key to communicate in a way that invites feedback and provides time to

alter course if necessary. If the position of your communication is “the decision has been made; there is no time to change,” then you aren’t communicating, you’re dictating. You’re preventing people who should be allowed to provide input into decisions from being able to do so. This isn’t a way to positively influence your organization or find optimal solutions.

■ **Note** Communicating and telling are entirely different acts.

Finally, as a program manager, I dislike the excuse of lack of time to make a higher-quality choice. I am forced to concede that by the time the project manager got around to making a decision, the options were limited and the time available to make the decision was small. But that doesn’t excuse the weeks and months leading up to that point, where the project manager could have been working on alternatives, communicating with stakeholders and team members to find an optimal solution, and preparing for the consequences of whatever decision they finally made. Yes, they put themselves in a spot where they had few options. But that was likely their own doing.

Doing things intentionally is a key to no-drama project management. Not only are decisions usually better if they’re made ahead of time and on purpose, but they also provide time for good communication and analysis, to ensure that you’re on the right track and doing the right thing. If you leave decisions to chance or wait until the last minute, you may find that your options are limited and your ability to find and discuss the best solutions have passed you by.

Expecting the Expected

This is another recurring theme in a no-drama management style: expecting the problems that you should have seen coming. With hundreds of thousands of project-management types managing over a million projects a year, certain patterns emerge that are fairly recognizable and predictable. Virtually all project managers have had to deal with team members who were shared with other projects or reassigned with little to no notice (and no replacement). You have to deal with client and stakeholder communities who can’t agree on the goals of the project, or who have extreme difficulty in determining priorities. You may face project drift caused by clients changing their mind or the market changing it for them. All of these occurrences are unfortunate, but

none of them are completely unforeseeable shocks that could never be predicted. Chapter 11 talked about the two things that make it possible to expect the unexpected: knowledge and strategy.

Although many of the problems you may face are predictable and can be expected, you need to know about them in order to prepare for them. If you can only rely on your own personal experience, then you're limited to the things you have had to handle yourself. This isn't just about learning best practices and studying project management. You need to talk candidly with your peers and with colleagues in the industry to gain an understanding of the problems you're likely to face and what the signs are, so you can best recognize them. It's very easy to become so wrapped up in your project or organization that you forget how many others like you are striving to do better and improve the discipline as a whole.

■ **Note** The greater project-management community is there to help.

But familiarity with consistent problems is only part of the task. You must learn how to recognize these problems as they're happening, and you must prepare to deal with them before they happen. If you don't recognize them and aren't ready for their occurrence, knowing about them doesn't do you much good. Some challenges are universal to all projects, like communication. Others are specific to your industry: problems that are common in software may not be prevalent in drug research. And finally, certain problems happen more frequently in your organization than anywhere else. As a good no-drama project manager, you need to be well-versed in the problems that may occur and the ways they're often solved.

The key is to not be surprised by problems that you could have predicted and avoided. Not all causes, and not all problems, can be foreseen, but you should be prepared for the problems and impacts that could have been. With this in hand, you can bring a measure of comfort to your team and client that will help keep the level of stress to a minimum.

Problems in Team Management

This book discussed five areas of team management that, if handled improperly, can lead to drama in your project. One aspect that makes team management slightly different than client management is the amount of influence

that you, the project manager, have in the outcome. Clients can change their minds on a whim and often believe they can write a check (either literally or figuratively) to cover the damage they do. Sometimes they're even correct. That won't work with your team.

A project manager needs to become highly skilled at these five main facets of team management: communication, planning, preparing, establishing metrics, and project roles. Even if you aren't a hands-on manager, but rather a manager of managers, these skills continue to be important. It makes little difference if it's a 1-person project or a 100-person project—as soon as the number of people on the project exceeds 1, these tenets become vital to the project's overall health.

Communication

One of the primary tasks of a project manager is managing communications with the entire project community, including the team, the client, the stakeholders, the service providers, and your own management chain. Through good communication, you can keep your team aligned and make sure they understand the ultimate goals of the project. And through alignment, you can produce significantly more value than you could if team members were pulling in opposite directions or otherwise weren't in sync with each other.

In addition to keeping your team aligned, you need to keep your clients aligned with each other as well as aligned with the team and the project. Often it's easier to do the latter than the former. The clients are often aware of what is being delivered, but they can't agree about the project goals. Although your primary responsibility may be to delivery, you should also spend time trying to get your clients on the same page. Otherwise you'll struggle to have a satisfied customer.

Your communications and team management will catch the eye of your program manager and probably the top levels of your client organization. Often, your communications are all they have to judge how well the project is being run. You need to consider this when you're structuring your communication plan. A certain group of people will only see your communications and never your delivery. With that in mind, what would you do differently?

It would be hard to overestimate the importance and value of project communications. It's important to keep your team headed in the same direction, and it's the only way to keep the client and team together. This is why I sometimes call communicating the primary job of the project manager.

Planning

Back in Chapter 10, I covered the importance of planning. To many experienced project managers, this seems obvious. How could you possibly have a project without also having a project plan?

It can be more of a challenge than you'd think. Planning takes time and can cause tension with clients and team members who want to get moving because they already know what they need to do. Figuring out where you want to go is important before you start going, lest you wind up some place you don't want to be.

Conversely, it's important for the project manager to remember that they're managing a project, not a project plan. If the plan is no longer useful or isn't driving the value you hoped to create, then continuing to follow the plan makes little sense. No project manager will win thanks and praise for following a plan perfectly but winding up with a deliverable that doesn't meet the business need.

■ **Note** It's important to know your destination as well as the alternate routes you can take to get there.

Therefore, planning is the intersection of knowing where you're going and flexibility in how you'll get there. Trying to move forward without a plan is likely to end up in a bad place, but so is sticking with a plan that isn't working. This is why it's called project management rather than project plan management.

Preparing

Chapter 11 discussed the concept of preparing and how it's different than planning. For the most part, you plan for things you know about. You know what needs to get done, roughly when it must be complete, and approximately what resources you need to apply in the process. You may build in contingencies based on things you know or experience with past projects. But either way, you're going off things you know or think you know.

Being prepared is the opposite. It means being ready for things that you don't know about but that you know will have an impact on the project. I talked about how there is no way to predict all things that can cause problems—the

list is too long. However, you can determine the impacts that would mess up the plan or the project, and begin preparing for them. It should be your goal to be prepared in case of emergencies, even if there is no way to guess correctly what will cause that emergency.

Being prepared, even in general terms, is a great way to avoid drama in your project. Two different mindsets go into preparation. The first involves adjusting your mindset from thinking about what will go right to thinking about what *could* go wrong. When you acknowledge that things may not go as planned, you'll be much less surprised when that turns out to be the case. Second, you wind up thinking about contingencies and alternate paths that you can break out when the need arises, or that give you practice for coming up with them when a real problem hits.

Establishing Metrics

I often position the topic of metrics to my teams and project managers as follows: "How will we know when we're winning?" This gets them thinking about the problem in the right way, rather than thinking more strictly about being measured. But however you think about it, if you don't know how to tell whether you're doing well, you probably shouldn't start.

Chapter 12 explored two kinds of metrics: metrics about what the project is delivering, and metrics about the project itself. When you're crafting metrics about what the project is supposed to produce, it's important to make sure that what you're measuring is aligned with the goals of the project and the organization. Additionally, it's vital that what you're measuring is actually measurable, or it will be ignored by the team and the client.

When it comes time to measure the project itself, you can use several standard techniques. The two I highlighted are project efficiency as measured by resource utilization, and project effectiveness as measured by an earned-value chart. Both of these standard measurements are very useful in determining the current health of the project and whether the project is trending in a good direction. No single view can tell program managers everything they need to know, but these two views indicate quite a bit.

Finally, I discussed the two measurements that matter in the long term. A few months or a few years after a project, not many people will remember if you improved a KPI by 2.7% or 2.9%. But they will remember whether they want to work with you again and whether they trust you to run another of their projects. Although this isn't a metric that measures the project or delivery, in the long run it may be the only evaluation that matters.

Project Roles

A no-drama project manager should consider three things when crafting a project team. The first is creating a team structure that is suitable for running the project successfully. You need a team that can both understand and deliver on the needs of the client. You also need team members capable of communicating with you and the other team members about their status and participating in the project to the level you need.

Second, you want to create a balanced team. This means having the right number of people in positions of leadership, and enough people who are being led. Depending on the size of the project, you may have technical leaders, quality leaders, user-experience leaders, or several of each. You should also consider the balance of skills, both technical and nontechnical. Make sure you have the expertise required to accomplish the project goals and also that you have a balance of styles and interests on the project. A good team is one in which the members complement each other, aren't identically skilled, and don't all have very similar outlooks and viewpoints.

■ **Note** A well-constructed team is balanced.

The last important thing to consider is setting clear expectations for your team members about their roles. Leaders need to know that you expect them to lead, and team members need to be aware of who is leading them. Similarly, skilled people must be aware of which skills they will use for the project and which they won't. A good source of project drama is having the wrong person doing a specific task and not doing it as well as someone else could. When team members are clear about the expectations of their roles, it's up to you to keep them performing by using everything from gentle reminders to direct conversation. You may be in an environment where anyone can do most anything, but that's not a particularly good way to run a project team.

Problems in Client Management

It's possible for a project's biggest failing to be the tactical management of day-to-day activities, but it's more likely that the problem arose earlier than that. A successful sports coach may leave the play calling to someone else but keep the leadership and motivational tasks for himself. Similarly, being

able to manage the client relationship with the project is critical not only to project success but also in the effort to limit the overall drama created by the project.

The client-management topics a project manager must excel at include understanding client needs and ensuring that clients are aligned with the project, with each other (if there is more than one), and with the decision-making process both before the project starts and while its running. Finally, the project manager must have a keen eye for stakeholder identification, because not everyone important to the project will be immediately apparent.

Requirements

At the base of every project is the list of items the client or sponsor wants done. However, many projects begin with that list, which can lead down the wrong path. A project isn't just a collection of client desires, with the ultimate hope that they will synthesize into something great. The project needs to start with a goal in mind, and the requirements collection and definition should flow from that. Your project almost certainly doesn't have an unlimited budget, so determining the key requirements helps decide how that budget should be spent.

Once you have a good idea what the project's critical requirements are, you should seek two kinds of approval. The first is the formal kind, where you attempt to get sign-off on what the project will deliver. This approval often feels more contractual than it does like an agreement between partners who want the same thing. In many ways, this is how it should feel, because formal sign-off is the mechanism used by vendors and third parties who want to understand what they need to deliver in order to fulfill their obligation. To get to this spot usually takes weeks or months, and often the result is a document full of compromises that everyone has agreed with at one point or another. This helps avoid contractual drama: when it comes time to settle the bill, it will be clear whether the project delivered as promised.

Contractual drama is easily remedied and isn't always meaningful to a company looking to form a long-term relationship. Sometimes you want to focus not on delivering features and functionality but on helping to drive long-term value. And this is where the second kind of sign-off comes into play: the "Do we still want to do this?" agreement. Once everyone has a good grasp of what the project includes and what it doesn't, do they still want to move forward? This is a question you should absolutely ask yourself, your team, and your client at regular intervals during the project. This implicit statement that the client likes the direction and wants to continue is valuable to all the

stakeholders involved. It's like checking the map when you're half way to your destination, to make sure you're headed to the right place. Nothing bad ever comes from double-checking.

Prioritization

If you gave your client a list of a dozen things that came up during requirements gathering, and asked which items they really wanted, the answer would be "All of them." This isn't because clients act like petulant children, it's because that's the wrong question. The client does want all 12 of those requirements. Otherwise they wouldn't have asked for them in the first place. The trick is to put the requirements in some kind of order.

Chapter 4 looked at two different ways to do this. First, you can group requirements into As, Bs, and Cs. Cs are things the project almost certainly won't do, and As are things that are most likely to get done. The list of Bs is where project management comes in. You commit to some of them, but not all; and you don't commit to any of them individually. It's a tricky balance, but one that I've seen work repeatedly. None of the Bs are more important than any of the As, and therefore, each of the Bs can be sacrificed for more important work. But you can say with a level of confidence that some of them will get done.

The second strategy is pure ranking of requirements: literally numbering them from one to whatever. Although this exercise becomes difficult and tedious the further down the list you go, the results toward the top are almost always incredibly interesting. By determining the top three or top ten requirements out of hundreds or thousands, you can do a much better job of focusing your efforts and your team's efforts on the right tasks. Without attempting this exercise, you'd never know.

■ **Note** Sometimes you learn a lot from the prioritization exercise itself.

Finally, you can also learn a great deal by determining what can be deprioritized. An awful lot of good ideas wind up on this list, which can be surprising. But just because something is a good idea, doesn't mean you should work on it ahead of other things, or work on it at all right now. Finding out this kind of information before you begin improves your planning and delivery.

Alignment

Your project won't be successful if the people and stakeholders on the project are all pulling in separate directions toward separate goals. This seems obvious, and it probably is obvious when people make completely opposite requests. But misalignment is a lot harder to spot. People talk in generalities, or they mostly agree, or the areas of disagreement don't seem material. You know everyone isn't perfectly aligned, but you think you have enough agreement to move forward. This happens all the time.

Chapter 6 talked about checking project alignment by using project principles. By distilling the hundreds of hours and thousands of pages of work already done into a dozen or fewer assertions about the project, you can do more to get agreement or foster dissent than you can by letting everyone read your huge pile of documentation. If your principles are clear enough, and you state firmly what you're attempting with the project, the result is discussion and constructive arguments. In the end, this is what you want. Getting everything out in the open is good, especially before it's too late to change course.

Alignment over project goals is paramount to running a good project. However, alignment over features, requirements, or tasks isn't. You should expect constant discussion over those, while your overall goals remain the same. If you have the opposite situation—alignment about tasks but not goals—then you appear to have agreement when in reality you have almost none.

Assumptions

Projects can't be run without assumptions. In fact, nothing would run if we were unable to make assumptions, most of which turn out to be true. Because projects are often approved and started with incomplete, imperfect information, you're forced to use assumptions to make up the difference. This is not only normal, it's the only way to move forward.

Assumptions can sometimes be tricky to spot, because they generally seem to be true to all involved; you would never attempt to assume something that is obviously false. Watch for the use of the vocabulary of assumptions. People talk about being “pretty sure” or being mostly confident. This is especially true when what they're talking about isn't in their area of expertise. The more someone talks definitively about something outside of their expertise, the more likely they are to be making an assumption.

Again, making assumptions is not only okay, it's expected for a project. However, once you identify the assumptions, you need to track them and decide if they actually are assumptions, or if they are risks, or something that you can manage. By the time the project is coming to a close, it's no longer okay to be working from assumptions; one of your tasks is to figure out whether those assumptions were true. Some of them will become risks; many others will be retired. You can only know if you manage them actively.

Stakeholder Identification

Making sure you know who your stakeholders are goes a long way toward reducing drama in your project. Some of them are easy to spot, such as the direct client, or the person paying the bill. These people have a direct impact on the project and are the ones from whom you're correct to take direction. They also include people who have a lot of interest in the project for various reasons.

■ **Note** Your project probably has more stakeholders than you realize.

Often, there is another class of stakeholder that is much harder to uncover. These are people who have heavy influence on the outcome, even though you can't find them on a project org chart or identify them ahead of time. Some of them want the project to succeed; some of them don't. But none of them can be safely ignored. You need to know who they are and what their hopes for the project are, positive and negative. Only then can you hope to manage them as stakeholders.

Managing the Unexpected

The last thing I talked about in Chapter 14 was the idea that despite all your planning, preparation, and careful thinking, things may happen that you didn't expect. Not only didn't you expect them, no one could have. If all the problems you face were foreseeable, no one would need you to manage projects, so you should probably be thankful. It's one of the things that make the job interesting, too.

I talked about how preparation factors in. Although it's impossible to be prepared for the unexpected, probably by definition, sometimes you can prepare

for the impacts of unforeseen events. Few of the causes are predictable, but their effects may be. It's possible that you don't need to add drama to your project because you've already thought of this problem, or a problem just like it, and have a solution in mind. If so, this is a great outcome.

On the other hand, if you're facing a problem that you couldn't anticipate and for which you have no ready solution, then you generally have three options. You can restart the project, you can revise the project, or you can rescope it. Each has its use, depending on the situation, and each has a certain tax that must be paid. But there is no sense moving forward with a project that has hit an unforeseen problem that you can't overcome. In the long run, that's wasted effort and a way to speed along the road to drama.

Your goal is to end the drama in your projects. Over the course of your career, you'll be responsible for running many projects with varied success. Sometimes project managers have spectacular failures. Other times, they have successes that are written about in newspapers and magazines. But above all, you should focus on delivering value to your clients while reducing the anxiety and stress of the job. That's what a No-Drama Project Manager finds most satisfying.

I

Index

A

- above the line (A-priority)
 - prioritization, 46
- active deprioritization, 54
- aggregation projects. *See* bottom-up requirements gathering
- aggressiveness, amount of, 86–87
- agreement, lack of high-level, 160–162
- alignment
 - around corporate or organizational strategy, 22
 - around goals of project, 5, 22
 - with client, 75–88
 - aligning with team, 80–82
 - amount of aggressiveness, 86–87
 - communicating alignment, 82–84
 - differences of opinion, 84–86
 - importance of, 76–77
 - remaining aligned, 87
 - expected problems in client management, 22
 - problems in client management, 237
- analyst role, 196–199
- analysts, types to be wary of, 198–199
- Analytic Hierarchy Process (pairwise prioritization), 52–54
- approval of projects, 235–236
- assertions, 92
- assumption lifecycle, 94
- assumptions, 89–104
 - avoiding standard or implicit, 98–99
 - definition of, 90–92
 - expected problems in client management, 23
 - identifying, 94–98
 - assumptions made by project team, 96
 - level of confidence, 97–98
 - pre-project assumptions, 94–96
 - importance of, 92–94
 - problems in client management, 237–238
 - vs. risk, 99–101
 - testing and tracking, along with risks, 101–102
 - wrong, 102–103
- audiences, communication for, 126–132
 - challenges of, 133–134
 - clients, 130–132
 - executives, 130–132
 - extended team, 128–129
 - internal team, 126–127
 - other departments, 129–130

B

- balanced teams, creating, 234
- below the line (C-priority)
 - prioritization, 46, 54–55
- benefits, of communication, 139–140
- bills, identifying decision makers by
 - payment of, 107–109
- bottom-up requirements gathering, 33
- bottom-up thinking, top-down thinking vs., 33–35
- broadcast communications, 136
- burndown charts, 181
- business owners (product managers),
 - project sponsors or clients versus, 12

C

- C-priority (below the line)
 - prioritization, 46, 54–55
- cause of problems, segregating from solution, 218
- challenges, of communication, 133–134
- change management, 59–74
 - communication and, 25
 - expected problems in client management, 21–22
 - importance of, 61–62
 - keeping focused, 73
 - people involved, 71–72
 - project implications, 72
 - reasons for, 62–69
 - failure in execution, 69
 - new legislation, 67–68
 - new sales channels, 66–67
 - organization has changed, 63–64
 - priorities have changed, 65–66
 - project affected by others, 68
 - project becomes obsolete, 64–65
 - situation caused by, 69–71
- change order, 214
- channels, for communication, 136–138
 - broadcast communications, 136
 - closed-circuit communication, 137–138
 - individual communications, 137
- Churchill, Winston, 59
- churn, subscription-based project, 56
- clarity around tasks and subtasks, 5
- client interface role, 201–202
- client management
 - expected problems in, 19–24
 - alignment, 22
 - assumptions, 23
 - change, 21–22
 - identifying stakeholders, 23–24
 - prioritization, 21
 - requirements, 20
 - problems in, 234–238
 - alignment, 237
 - assumptions, 237–238
 - prioritization, 236
 - requirements, 235–236
 - stakeholder identification, 238
- client satisfaction, metrics, 183–184
- clients
 - alignment with, 75–88
 - amount of aggressiveness, 86–87
 - communicating, 82–84
 - differences of opinion, 84–86
 - importance of, 76–77
 - remaining aligned, 87
 - and team, 80–82
 - communication, 130–132, 231
 - role of, 12–13
- closed-circuit communication, 137–138
- code words, 83
- coincidental action, intentional action versus, 145
- communication, 123–141
 - alignment, 82–84
 - benefits of, 139–140

- of changes, 62, 70
 - channels for, 136–138
 - broadcast communications, 136
 - closed-circuit communication, 137–138
 - individual communications, 137
 - for different audiences, 126–132
 - challenges of, 133–134
 - clients, 130–132
 - executives, 130–132
 - extended team, 128–129
 - internal team, 126–127
 - other departments, 129–130
 - effectiveness of, 138–139
 - expected problems in team management, 24–25
 - importance of, 125–126
 - out-of-channel escalations, 18–19
 - problems in team management, 231
 - as project, 134–136
 - role of project manager, 6
 - strategy for, 38–40
 - with team about unexpected events, 210
 - treating like deliverable, 227–228
 - communication plan, 152
 - completed project, 49
 - complexity, underestimating, 165–167
 - communication, medium versus message, 125
 - confidence
 - handling wrong assumptions, 102–103
 - level of, 97–98
 - constraints
 - based on out-of-date assumptions, 96
 - requirements versus, 36
 - consultant, assumptions and, 91
 - contractual drama, 235
 - core team, 192
 - creativity, of project manager, 214
 - critical assumptions, 92
 - critical path, 49–50
 - critical requirements, 47
 - customers
 - alignment with, 77
 - communication with, 231
 - identifying, 31–33, 112–114
 - requirements gathering from, 32–33
 - types of, 31
- ## D
- data, basing KPIs on valid, 176–178
 - decision makers, identifying, 105–121
 - customers, 112–114
 - influencers, 116–117
 - managing stakeholders, 118–121
 - by payment of bills, 107–109
 - by presence of human resources, 109–112
 - signature authority, 114–116
 - decisions, intentional, 228–229
 - delivering solutions, 113–114
 - departments, communication for, 129–130
 - dependency chain, critical path for project, 49
 - deprioritizing
 - proper procedure for, 55–56
 - value of, 54–55
 - developer role, 199–200
 - drama, 223–239
 - avoiding, 224–228
 - being open and honest, 225–227
 - preparation, 224–225
 - with preparation, 233
 - treating communication like deliverable, 227–228
 - contractual, 235
 - due to unexpected problems, 210
 - making intentional decisions, 228–229

drama (*continued*)
problems
 in client management, 234–238
 expecting foreseeable, 229–230
 managing unexpected, 238–239
 in team management, 230–234

E

earned value, metrics, 181–182
effectiveness, of communication,
 138–139
effort time versus actual time, 190
elevator pitch, 39
errors
 avoiding repetition of, 223
 learning from, 217–218
executives, communicating with,
 133–134
 expectations, 131
 status, risk, and progress
 updates, 131
 what they can do to help, 131
executives, communication for,
 130–132
expectations of roles in teams, 234
expecting and preparing for problems,
 207–208
experience, learning to deal with
 unexpected events, 206
extended team, communicating
 with, 133
 project status, 129
 requirements, deadlines, and
 deliverables, 129
 value of the work, 128
extended team, communication for,
 128–129

F

failure, 15–16
 in execution, 69

 identifying those hoping for,
 117–118
 at obvious, 17–18
false information, in plans, 147–148
feedback, inviting, 228
final approval, importance of
 determining who has, 116
finance tool, decision makers
 regarding, 107
focus
 keeping through changes, 73
 value of, 37–38
formal sign-offs, 40–41
functional manager role, 195–196

G

gap assignments, 3
go-forward team, 220
goals
 aligning KPIs with organization,
 173–174
 driving right behavior
 through, 175
Golden Rule, decision making, 109
Gray, John, 34
group project managers. *See* program
 managers
Groupon, 67–68
guesses, versus assumptions, 91

H

handling wrong assumptions, 102–103
honesty, avoiding drama through,
 225–227
HRIS (Human Resources Information
 System), decision makers
 regarding, 107
human resources, identifying decision
 makers by presence of, 109–112

Human Resources Information System (HRIS), decision makers regarding, 107

I, J

identifying, decision makers, 105–121
 customers, 112–114
 influencers, 116–117
 managing stakeholders, 118–121
 by payment of bills, 107–109
 by presence of human resources, 109–112
 signature authority, 114–116

implicit assumptions, avoiding, 98–99

importance
 of change management, 61–62
 of communication, 125–126

individual communications, 137

inflexible plans, 148–149

influence/interest matrix, 119–121

influencers
 decision making, 116–117
 identifying, 116–117

informal sign-offs, 40–42

information sharing, communication versus, 124

intentional decisions, 145–146, 153, 228–229

interdepartmental differences, 84

internal (project) team,
 communicating with, 126–127, 133–136, 139–140
 decisions and changes, 127
 project updates, 127
 roles, responsibilities, and goals of, 127
 task coordination, 127

internal tool, decision makers regarding, 107

interview questions
 aligning with client, 78–79

aligning with team members, 81–82

intradepartmental differences, 85

K

KPIs (Key Performance Indicators), 173–178
 aligning with organization goals, 173–174
 basing on valid data, 176–178
 driving correct behavior with, 175–176
 establishing, 26–27

L

laundry list projects. See bottom-up requirements gathering

leaders, 191–192

legislation, changes caused by, 67–68

level of confidence, 97–98

levels, of team members, 191–194
 leaders, 191–192
 team contributors, 192–194
 team members, 192

M

management
 change management, 59–74
 communication and, 25
 expected problems in client management, 21–22
 importance of, 61–62
 keeping focused, 73
 people involved, 71–72
 project implications, 72
 reasons for, 62–69
 situation caused by, 69–71

client management
 expected problems in, 19–24
 problems in, 234–238

project-management principles, 86–87

risk-management matrix, 151

management (*continued*)
 risk-management plans, 150–152
 team management
 alignment, 80–82
 expected problems in, 24–28
 identifying assumptions, 96
 problems in, 230–234

market research, assumptions and, 91

market risk, 72

material change, 63–64

material changes to projects, 215

McCain, Brad, 139

metrics, 171–186
 client satisfaction, 183–184
 earned value, 181–182
 establishing
 expected problems in team management, 26–27
 problems in team management, 233
 KPIs, 173–178
 aligning with organization goals, 173–174
 basing on valid data, 176–178
 driving correct behavior with, 175–176
 resources, 179–180
 team satisfaction, 184–185
 top-level, 172–173
 visions versus, 177

micromanaging, assumptions, 99

mistakes
 avoiding repetition of, 228
 avoiding repetition of, 223
 learning from, 217–218

mitigatable assumptions, 100

mitigatable risk, 100

morale, team, alignment and, 76

N

no-drama preparation, 224

nonfunctional requirements, 36

O

observer effect, 117

opinions, differences of, 84–86

organization
 aligning KPIs with goals of, 173–174
 changes, 63–64

organizational change, 63

out-of-channel escalations, 18–19

outside-in requirements gathering, 35–37

overplanning, 146–147

P, Q

pairwise prioritization, 51–54

people, involved in changes, 71–72

performance, team, 76

personal ROI, 175

Phase 2, naming rescoped projects, 216

plan development, proceeding without, 143–145

planning, 143–154
 acting intentionally, 145–146
 communication plan, 152
 expected problems in team management, 25
 false information in, 147–148
 inflexible, 148–149
 overplanning, 146–147
 preparing versus, 25
 problems in team management, 232
 risk-management plans, 150–152
 staffing plans, 149–150
 sticking with, 153–154

PMBOK (Project Management Body of Knowledge), documenting assumptions, 89

PMI (Project Management Institute), number certified through, 1

- PMO (Project Management Office), 10–11
- policing, roles, 202–203
- portfolio managers. *See* program managers
- pre-project assumptions, identifying, 94–96
- preparation
 - avoiding drama through, 224–225
 - for impacts of unexpected events, 238
- preparing
 - expected problems in team management, 25–26
 - problems in team management, 232–233
- prioritization, 45–57
 - changes in, 65–66
 - critical path, 49–50
 - deprioritizing
 - proper procedure for, 55–56
 - value of, 54–55
 - expected problems in client management, 21
 - importance of, 47–49
 - pairwise, 51–54
 - playing games with, 50–51
 - problems in client management, 236
 - of projects, effect of restarting projects on, 211
- problem statement, assumptions and, 93
- problems, 155–169
 - breaking process, 163–165
 - in client management, 234–238
 - alignment, 237
 - assumptions, 237–238
 - prioritization, 236
 - requirements, 235–236
 - stakeholder identification, 238
 - expected
 - in client management, 19–24
 - in team management, 24–28
 - expecting foreseeable, 229–230
 - lack of high-level agreement, 160–162
 - schedule “make up”, 167–168
 - skill gaps, 162–163
 - slow turnaround, 158–160
 - in team management, 230–234
 - communication, 231
 - establishing metrics, 233
 - planning, 232
 - preparing, 232–233
 - project roles, 234
 - unclear project ownership, 157–158
 - underestimating complexity, 165–167
 - unexpected, 28, 205–221, 238–239
 - fallout from, 217–219
 - managing team, 219–220
 - preparation for, 207–208
 - strategies for, 210–213, 216–217
- process, breaking, 163–165
- product managers (business owners), project sponsors or clients versus, 12
- profitability, 172
- program managers
 - expected problems, 156
 - failure and, 15–16
 - role of, 8–10
 - slow turnaround, 159
 - unclear project ownership, 158
- program sponsors or clients
 - awareness of effects of change on, 71
 - client-contact role, 201–202
 - communication with, 25, 131, 133–134, 139–140
 - lack of alignment with, 17
 - measuring satisfaction of, 184–185
 - sign-off follow-up, 42
 - slow turnaround, 159–160
 - unclear project ownership, 157–158

project change, 60. *See also* change management

Project Management Body of Knowledge (PMBOK), documenting assumptions, 89

Project Management Institute (PMI), number certified through, 1

Project Management Office (PMO), 10–11

project-management principles, 86–87

project managers

- motivations for, 2–3
- program managers versus, 9
- Project Management Office and, 11
- project sponsors or clients and, 12–13
- role of, 5–6
- weaknesses of
 - false information, 147–148
 - inflexibility, 148–149
 - overplanning, 146–147

project principles, 83–84

project roles, problems in team management, 234

project sponsors, role of, 12–13

project team, prioritization exercise, 21

project teams

- alignment with, 80–82
- identifying assumptions made by, 96
- role of, 7–8

projects

- affected by other projects, 68
- approval of, 235–236
- becomes obsolete, 64–65
- benefits of managing, 2–3
- communication as, 134–136
- honesty with status of, 225–227
- implications on when there are changes, 72
- metrics, 233
- strategy pyramid, 77–78
- unclear ownership of, 157–158

R

ranking of project requirements, 236

reasons, for change, 62–69

- failure in execution, 69
- new legislation, 67–68
- new sales channels, 66–67
- organization has changed, 63–64
- priorities have changed, 65–66
- project affected by others, 68
- project becomes obsolete, 64–65

requirements

- expected problems in client management, 20
- gathering
 - client management and, 20
 - communication and, 25
 - focus on system and tactical requirements, 30
 - incompleteness of, 29
- identifying, 29–44
 - communication strategy, 38–40
 - customers, 31–33
 - outside-in requirements gathering, 35–37
 - sign-off, 40–43
 - top-down thinking vs. bottom-up thinking, 33–35
 - value of focus, 37–38
- problems in client management, 235–236
- project, pairwise prioritization and, 54

resource owner, decision inputs, 110–112

resource risk, 72

resources, metrics, 179–180

return on investment. *See* ROI

reversal of decisions, 158

risk analysis, change management and, 72

risk-management matrix, 151

risk-management plans, 150–152

risk minimizers, 115–116
 risk, turning assumptions into, 94
 risks, and assumptions
 risks vs., 99–101
 testing and tracking, 101–102
 ROI (return on investment)
 communication and, 124–125
 effect of rescoping projects
 on, 216
 effect of restarting projects
 on, 211
 KPIs and, 174
 program managers and, 9–10

roles

analyst, 196–199
 client interface, 201–202
 defining, 194–195
 developer, 199–200
 expectations of in teams, 234
 functional manager, 195–196
 necessity of, 3–4
 of PMO, 10–11
 policing, 202–203
 program manager, 5–6, 8–10
 project
 expected problems in team
 management, 27–28
 problems in team
 management, 234
 project sponsor and/or client,
 12–13
 project team, 7–8
 tester/reviewer, 200–201

Rumsfeld, Don, 16

S

sales channels, changes caused by,
 66–67
 schedule, “making up” when behind,
 167–168
 schedule risk, 72
 scope, 181

sign-offs, 40–43
 following up on, 42–43
 formal, 40–41
 informal, 41–42
 signature authorities, identifying,
 114–116
 sine qua non concept, 50
 situation, caused by changes, 69–71
 skill gaps, 162–163
 skills, team management, 232
 solution of problems, segregating
 from cause, 218
 sponsors, role of, 12–13
 staffing plans, 149–150
 stakeholder identification, problems in
 client management, 238
 stakeholders. *See also* decision makers
 alignment, 22
 communicating with, 135, 152
 as decision makers, 105–107
 identifying, expected problems in
 client management, 23–24
 managing, 118–121
 standard assumptions, avoiding, 98–99
 status updates on projects, 225–227
 strategic direction, strategy pyramid,
 77–78
 strategy pyramid, 77–78
 stress, causing for project manager,
 18–19
 subject-matter expert, assumptions
 and, 91
 subscription churn, 177–178
 success, 15–28
 being headache, 18–19
 criteria, managing
 intradepartmental
 differences, 86
 expected problems
 in client management, 19–24
 in team management, 24–28

- success (*continued*)
 - failing at obvious, 17–18
 - unexpected problems, 28
- support of team, 226
- T**
- tasks, strategy pyramid, 78
- tax laws, 67–68
- team management
 - alignment, 80–82
 - expected problems in, 24–28
 - communication, 24–25
 - establishing metrics, 26–27
 - planning, 25
 - preparing, 25–26
 - project roles, 27–28
 - identifying assumptions, 96
 - problems in, 230–234
 - communication, 231, 234
- teams, 187–204
 - benefits of being honest with, 225–227
 - benefits of being open with, 225
 - effect of restarting projects on, 212
 - effect of revising projects on, 214–215
 - and group leaders
 - making good use of, 7
 - underestimating complexity, 166
 - understanding leadership role of, 28
 - and group members
 - alignment and, 76
 - awareness of effects of change on, 71–74
 - expressing status, 190
 - measuring satisfaction of, 183–184
 - proper execution, 189–190
 - roles of, 8
 - skill gap, 162–163
 - team contributors versus, 193–194
 - understanding needs and objectives of project, 189
 - importance of communication with, 231
 - keeping from worrying about unexpected events, 210
 - managing, when facing unexpected problems, 219–220
 - member levels, 191–194
 - leaders, 191–192
 - team contributors, 192–194
 - team members, 192
 - necessary abilities for
 - executing on tasks, 189–190
 - expressing status, 190
 - understanding project, 189
 - roles in
 - analyst, 196–199
 - client interface, 201–202
 - defining, 194–195
 - developer, 199–200
 - functional manager, 195–196
 - policing, 202–203
 - tester/reviewer, 200–201
 - satisfaction, metrics, 184–185
- technological change, 64–65
- tester/reviewer role, 200–201
- those hoping for failure, 117–118
- top-down thinking, vs. bottom-up thinking, 33–35
- top-level metrics, 172–173
- training peers, to deal with unexpected problems, 206
- transparency in decision-making, 61, 73
- troubleshooting, 155–169
 - breaking process, 163–165
 - in client management, 234–238
 - alignment, 237
 - assumptions, 237–238
 - prioritization, 236
 - requirements, 235–236
 - stakeholder identification, 238
 - expected problems
 - in client management, 19–24
 - in team management, 24–28

- expecting foreseeable problems, 229–230
- lack of high-level agreement, 160–162
- schedule “make up”, 167–168
- skill gaps, 162–163
- slow turnaround, 158–160
- in team management, 230–234
 - communication, 231
 - establishing metrics, 233
 - planning, 232
 - preparing, 232–233
 - project roles, 234
- unclear project ownership, 157–158
- underestimating complexity, 165–167
- unexpected problems, 28, 205–221, 238–239
 - fallout from, 217–219
 - managing team, 219–220
 - preparation for, 207–208
 - strategies for, 210–213, 216–217
- truth, importance of, 225–227
- turnaround, slow, 158–160
- Two-level prioritization, 46

U

- underestimating complexity, 165–167
- unexpected problems, 238–239. *See also* problems
- unknown, managing, 94
- unmitigatable assumptions, 100
- unmitigatable risk, 100
- usability testing, 214–215

V

- value of project revisions, understanding, 214

W, X

- wrong assumptions, handling, 102–103

Y, Z

- Y2K problem, 36–37