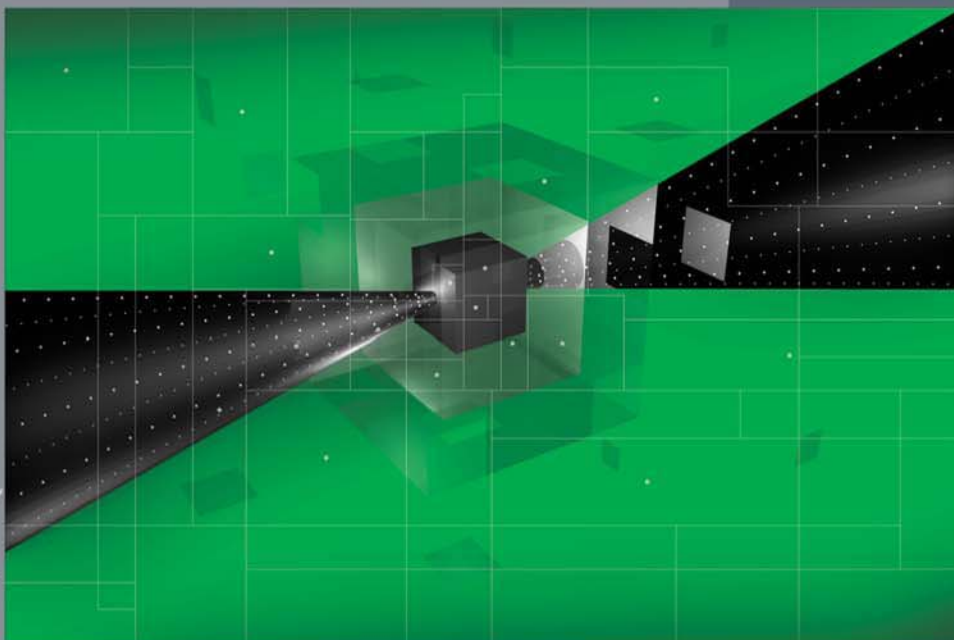
Microsoft

BEST PRACTICES

SOFTWARE ESTIMATION



Demystifying the Black Art

Steve McConnell

Two-time winner of *Software Development* magazine's Jolt Award

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number 2005936847

ISBN: 978-0-7356-0535-0

Printed and bound in the United States of America.

7 8 9 10 11 12 13 14 15 QGT 7 6 5 4 3 2

Distributed in Canada by H.B. Fenn and Company Ltd..

A CIP catalogue record for this book is available from the British Library

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to mspinput@microsoft.com.

Acquisitions Editor: Ben Ryan
Project Editor: Devon Musgrave
Copy Editor: Becka McKay
Indexer: Seth Maislin

Body Part No. X11-82276

Contents at a Glance

Table of Contents

Part II Fundamental Estimation Techniques

Welcome

The most unsuccessful three years in the education of cost estimators appears to be fifth-grade arithmetic.

–Norman R. Augustine

Software estimation is not hard. Experts have been researching and writing about software estimation for four decades, and they have developed numerous techniques that support accurate software estimates. Creating accurate estimates is straightforward—once you understand how to create them. But not all estimation practices are intuitively obvious, and even smart people won't discover all the good practices on their own. The fact that someone is an expert developer doesn't make that person an expert estimator.

Numerous aspects of estimation are not what they seem. Many so-called estimation problems arise from misunderstanding what an "estimate" is or blurring other similar-but-not-identical concepts with estimation. Some estimation practices that seem intuitively useful don't produce accurate results. Complex formulas sometimes do more harm than good, and some deceptively simple practices produce surprisingly accurate results.

This book distills four decades of research and even more decades of hands-on experience to help developers, leads, testers, and managers become effective estimators. Learning about software estimation turns out to be generally useful because the influences that affect software estimates are the influences that affect software development itself.

Art vs. Science of Software Estimation

Software estimation research is currently focused on improving estimation techniques so that sophisticated organizations can achieve project results within ±5% of estimated results instead of within ±10%. These techniques are mathematically intensive. Understanding them requires a strong math background and concentrated study. Using them requires number crunching well beyond what you can do on your hand calculator. These techniques work best when embodied in commercial software estimation tools. I refer to this set of practices as the *science of estimation*.

Meanwhile, the typical software organization is not struggling to improve its estimates from ±10% to ±5% accuracy. The typical software organization is struggling to avoid estimates that are incorrect by 100% or more. (The reasons for this are manifold and will be discussed in detail in Chapters 3 and 4.)

Our natural tendency is to believe that complex formulas like this:

$$\text{Effort} = 2.94 * (\text{KSLOC})^{[0.91 + 0.01 * \Sigma_{j=1}^{5} SF_j]} * \Pi_{i=1}^{17} \text{EM}_i$$

will always produce more accurate results than simple formulas like this:

Effort = NumberOfRequirements * AverageEffortPerRequirement

But complex formulas aren't necessarily better. Software projects are influenced by numerous factors that undermine many of the assumptions contained in the complex formulas of the science of estimation. Those dynamics will be explained later in this book. Moreover, most software practitioners have neither the time nor the inclination to learn the intensive math required to understand the science of estimation.

Consequently, this book emphasizes rules of thumb, procedures, and simple formulas that are highly effective and understandable to practicing software professionals. These techniques will not produce estimates that are accurate to within ±5%, but they will reduce estimation error to about 25% or less, which turns out to be about as useful as most projects need, anyway. I call this set of techniques the *art of estimation*.

This book draws from both the art and science of software estimation, but its focus is on software estimation as an art.

Why This Book Was Written and Who It Is For

The literature on software estimation is widely scattered. Researchers have published hundreds of articles, and many of them are useful. But the typical practitioner doesn't have time to track down dozens of papers from obscure technical journals. A few previous books have described the science of estimation. Those books are 800–1000 pages long, require a good math background, and are targeted mainly at professional estimators—consultants or specialists who estimate large projects and do so frequently.

I wrote this book for developers, leads, testers, and managers who need to create estimates occasionally as one of their many job responsibilities. I believe that most practitioners want to improve the accuracy of their estimates but don't have the time to obtain a Ph.D. in software estimation. These practitioners struggle with practical issues like how to deal with the politics that surround the estimate, how to present an estimate so that it will actually be accepted, and how to avoid having someone change your estimate arbitrarily. If you are in this category, this book was written for you.

The techniques in this book apply to Internet and intranet development, embedded software, shrink-wrapped software, business systems, new development, legacy systems, large projects, small projects—essentially, to estimates for all kinds of software.

Key Benefits Of This Book

By focusing on the art of estimation, this book provides numerous important estimation insights:

- What an "estimate" is. (You might think you already know what an estimate is, but common usages of the term are inaccurate in ways that undermine effective estimation.)
- The specific factors that have made your past estimates less accurate than they could have been.
- Ways to distinguish a good estimate from a poor one.
- Numerous techniques that will allow *you personally* to create good estimates.
- Several techniques you can use to help *other people on your team* create good estimates.
- Ways that *your organization* can create good estimates. (There are important differences between personal techniques, group techniques, and organizational techniques.)
- Estimation approaches that work on agile projects, and approaches that work on traditional, sequential (plan-driven) projects.
- Estimation approaches that work on small projects and approaches that work on large projects.
- How to navigate the shark-infested political waters that often surround software estimation.

In addition to gaining a better understanding of estimation concepts, the practices in this book will help you estimate numerous specific attributes of software projects, including:

- New development work, including schedule, effort, and cost
- Schedule, effort, and cost of legacy systems work
- How many features you can deliver within a specific development iteration
- The amount of functionality you can deliver for a whole project when schedule and team size are fixed
- Proportions of different software development activities needed, including how much management work, requirements, construction, testing, and defect correction will be needed
- Planning parameters, such as tradeoffs between cost and schedule, best team size, amount of contingency buffer, ratio of developers to testers, and so on

- Quality parameters, including time needed for defect correction work, defects that will remain in your software at release time, and other factors
- Practically anything else you want to estimate

In many cases, you'll be able to put this book's practices to use right away.

Most practitioners will not need to go any further than the concepts described in this book. But understanding the concepts in this book will lay enough groundwork that you'll be able to graduate to more mathematically intensive approaches later on, if you want to.

What This Book Is Not About

This book is not about how to estimate the very largest projects—more than 1 million lines of code, or more than 100 staff years. Very large projects should be estimated by professional estimators who have read the dozens of obscure journal articles, who have studied the 800–1000-page books, who are familiar with commercial estimation software, and who are as skilled in both the art and science of estimation.

Where to Start

Where you start will depend on what you want to get out of the book.

If you bought this book because you need to create an estimate right now... Begin with Chapter 1 ("What Is an "Estimate"?), and then move to Chapter 7 ("Count, Compute, Judge") and Chapter 8 ("Calibration and Historical Data"). After that, skim the tips in Chapters 10–20 to find the techniques that will be the most immediately useful to you. By the way, this book's tips are highlighted in the text and numbered, and all of the tips—118 total—are also collected in Appendix C, "Software Estimation Tips."

If you want to improve your personal estimation skills, if you want to improve your organization's estimation track record, or if you're looking for a better understanding of software estimation in general... You can read the whole book. If you like to understand general principles before you dive into the details, read the book in order. If you like to see the details first and then draw general conclusions from the details, you can start with Chapter 1, read Chapters 7 through 23, and then go back and read the earlier chapters that you skipped.

Bellevue, Washington
New Year's Day, 2006

Microsoft Press Support

Every effort has been made to ensure the accuracy of this book. Microsoft Press provides corrections for books through the World Wide Web at the following address:

http://www.microsoft.com/learning/support/books/

To connect directly to the Microsoft Press Knowledge Base and enter a query regarding a question or issue that you may have, go to:

http://www.microsoft.com/mspress/support/search.asp

If you have comments, questions, or ideas regarding this book, please send them to Microsoft Press using either of the following methods:

Postal Mail:

Microsoft Press
Attn: Software Estimation *Editor*
One Microsoft Way
Redmond, WA 98052-6399

E-Mail:

mspinput@microsoft.com

Acknowledgments

I continue to be amazed at the many ways the Internet supports high-quality work. My first book's manuscript was reviewed almost entirely by people who lived within 50 miles of me. This book's manuscript included reviewers from Argentina, Australia, Canada, Denmark, England, Germany, Iceland, The Netherlands, Northern Ireland, Japan, Scotland, Spain, and the United States. The book has benefited enormously from these reviews.

Thanks first to the people who contributed review comments on significant portions of the book: Fernando Berzal, Steven Black, David E. Burgess, Stella M. Burns, Gavin Burrows, Dale Campbell, Robert A. Clinkenbeard, Bob Corrick, Brian Donaldson, Jason Hills, William Horn, Carl J Krzystofczyk, Jeffrey D. Moser, Thomas Oswald, Alan M. Pinder, Jon Price, Kathy Rhode, Simon Robbie, Edmund Schweppe, Gerald Simon, Creig R. Smith, Linda Taylor, and Bernd Viefhues.

Thanks also to the people who reviewed selected portions of the book: Lisa M. Adams, Hákon Ágústsson, Bryon Baker, Tina Coleman, Chris Crawford, Dominic Cronin, Jerry Deville, Conrado Estol, Eric Freeman, Hideo Fukumori, C. Dale Hildebrandt, Barbara Hitchings, Jim Holmes, Rick Hower, Kevin Hutchison, Finnur Hrafn Jonsson, Aaron Kiander, Mehmet Kerem Kızıltunç, Selimir Kustudic, Molly J. Mahai, Steve Mattingly, Joe Nicholas, Al Noel, David O'Donoghue, Sheldon Porcina, David J. Preston, Daniel Read, David Spokane, Janco Tanis, Ben Tilly, and Wendy Wilhelm.

I'd especially like to acknowledge Construx's estimation seminar instructors. After years of stimulating discussions, it's often impossible to tell which ideas originated with me and which originated with them. Thanks to Earl Beede, Gregg Boer, Matt Peloquin, Pamela Perrott, and Steve Tockey.

This book focuses on estimation as an art, and this book's simplifications were made possible by researchers who have spent decades clarifying estimation as a science. My heartfelt appreciation to three of the giants of estimation science: Barry Boehm, Capers Jones, and Lawrence Putnam.

Working with Devon Musgrave, project editor for this book, has once again been a privilege. Thanks, Devon! Becka McKay, assistant editor, also improved my original manuscript in countless ways. Thanks also to the rest of the Microsoft Press staff, including Patricia Bradbury, Carl Diltz, Tracey Freel, Jessie Good, Patricia Masserman, Joel Panchot, and Sandi Resnick. And thanks to indexer Seth Maislin.

Thanks finally to my wife, Ashlie, who is—in my estimation—the best life partner I could ever hope for.

Equations

Figures

Chapter 1

What Is an "Estimate"?

> *It is very difficult to make a vigorous, plausible, and job-risking defense of an estimate that is derived by no quantitative method, supported by little data, and certified chiefly by the hunches of the managers.*
>
> *–Fred Brooks*

You might think you already know what an estimate is. My goal by the end of this chapter is to convince you that an estimate is different from what most people think. A *good* estimate is even more different.

Here is a dictionary definition of *estimate*: 1. A tentative evaluation or rough calculation. 2. A preliminary calculation of the cost of a project. 3. A judgment based upon one's impressions; opinion. (Source: *The American Heritage Dictionary*, Second College Edition, 1985.)

Does this sound like what you are asked for when you're asked for an estimate? Are you asked for a *tentative* or *preliminary* calculation—that is, do you expect that you can change your mind later?

Probably not. When executives ask for an "estimate," they're often asking for a commitment or for a plan to meet a target. The distinctions between estimates, targets, and commitments are critical to understanding what an estimate is, what an estimate is not, and how to make your estimates better.

1.1 Estimates, Targets, and Commitments

Strictly speaking, the dictionary definition of *estimate* is correct: an estimate is a prediction of how long a project will take or how much it will cost. But estimation on software projects interplays with business targets, commitments, and control.

A *target* is a statement of a desirable business objective. Examples include the following:

- "We need to have Version 2.1 ready to demonstrate at a trade show in May."
- "We need to have this release stabilized in time for the holiday sales cycle."
- "These functions need to be completed by July 1 so that we'll be in compliance with government regulations."
- "We must limit the cost of the next release to $2 million, because that's the maximum budget we have for that release."

Businesses have important reasons to establish targets independent of software estimates. But the fact that a target is desirable or even mandatory does not necessarily mean that it is achievable.

While a target is a description of a desirable business objective, a *commitment* is a promise to deliver defined functionality at a specific level of quality by a certain date. A commitment can be the same as the estimate, or it can be more aggressive or more conservative than the estimate. In other words, do not assume that the commitment has to be the same as the estimate; it doesn't.

Tip #1 Distinguish between estimates, targets, and commitments.

1.2 Relationship Between Estimates and Plans

Estimation and planning are related topics, but estimation is not planning, and planning is not estimation. Estimation should be treated as an unbiased, analytical process; planning should be treated as a biased, goal-seeking process. With estimation, it's hazardous to want the estimate to come out to any particular answer. The goal is accuracy; the goal is not to seek a particular result. But the goal of planning is to seek a particular result. We deliberately (and appropriately) bias our plans to achieve specific outcomes. We plan specific means to reach a specific end.

Estimates form the foundation for the plans, but the plans don't have to be the same as the estimates. If the estimates are dramatically different from the targets, the project plans will need to recognize that gap and account for a high level of risk. If the estimates are close to the targets, then the plans can assume less risk.

Both estimation and planning are important, but the fundamental differences between the two activities mean that combining the two tends to lead to poor estimates *and* poor plans. The presence of a strong planning target can lead to substitution of the target for an analytically derived estimate; project members might even refer to the target as an "estimate," giving it a halo of objectivity that it doesn't deserve.

Here are examples of planning considerations that depend in part on accurate estimates:

- Creating a detailed schedule
- Identifying a project's critical path
- Creating a complete work breakdown structure
- Prioritizing functionality for delivery
- Breaking a project into iterations

Accurate estimates support better work in each of these areas (and Chapter 21, "Estimating Planning Parameters," goes into more detail on these topics).

1.3 Communicating about Estimates, Targets, and Commitments

One implication of the close and sometimes confusing relationship between estimation and planning is that project stakeholders sometimes miscommunicate about these activities. Here's an example of a typical miscommunication:

> *EXECUTIVE: How long do you think this project will take? We need to have this software ready in 3 months for a trade show. I can't give you any more team members, so you'll have to do the work with your current staff. Here's a list of the features we'll need.*
>
> *PROJECT LEAD: OK, let me crunch some numbers, and get back to you.*

Later…

> *PROJECT LEAD: We've estimated the project will take 5 months.*
>
> *EXECUTIVE: Five months!? Didn't you hear me? I said we needed to have this software ready in 3 months for a trade show!*

In this interaction, the project lead will probably walk away thinking that the executive is irrational, because he is asking for the team to deliver 5 months' worth of functionality in 3 months. The executive will walk away thinking that the project lead doesn't "get" the business reality, because he doesn't understand how important it is to be ready for the trade show in 3 months.

Note in this example that the executive was not really asking for an estimate; he was asking the project lead to come up with a *plan* to hit a *target*. Most executives don't have the technical background that would allow them to make fine distinctions between estimates, targets, commitments, and plans. So it becomes the technical leader's responsibility to translate the executive's request into more specific technical terms.

Here's a more productive way that the interaction could go:

> *EXECUTIVE: How long do you think this project will take? We need to have this software ready in 3 months for a trade show. I can't give you any more team members, so you'll have to do the work with your current staff. Here's a list of the features we'll need.*
>
> *PROJECT LEAD: Let me make sure I understand what you're asking for. Is it more important for us to deliver 100% of these features, or is it more important to have something ready for the trade show?*

EXECUTIVE: We have to have something ready for the trade show. We'd like to have 100% of those features if possible.

PROJECT LEAD: I want to be sure I follow through on your priorities as best I can. If it turns out that we can't deliver 100% of the features by the trade show, should we be ready to ship what we've got at trade show time, or should we plan to slip the ship date beyond the trade show?

EXECUTIVE: We have to have something for the trade show, so if push comes to shove, we have to ship something, even if it isn't 100% of what we want.

PROJECT LEAD: OK, I'll come up with a plan for delivering as many features as we can in the next 3 months.

Tip #2 When you're asked to provide an estimate, determine whether you're supposed to be estimating or figuring out how to hit a target.

1.4 Estimates as Probability Statements

If three-quarters of software projects overrun their estimates, the odds of any given software project completing on time and within budget are not 100%. Once we recognize that the odds of on-time completion are not 100%, an obvious question arises: "If the odds aren't 100%, what are they?" This is one of the central questions of software estimation.

Software estimates are routinely presented as single-point numbers, such as "This project will take 14 weeks." Such simplistic single-point estimates are meaningless because they don't include any indication of the probability associated with the single point. They imply a probability as shown in Figure 1-1—the only possible outcome is the single point given.

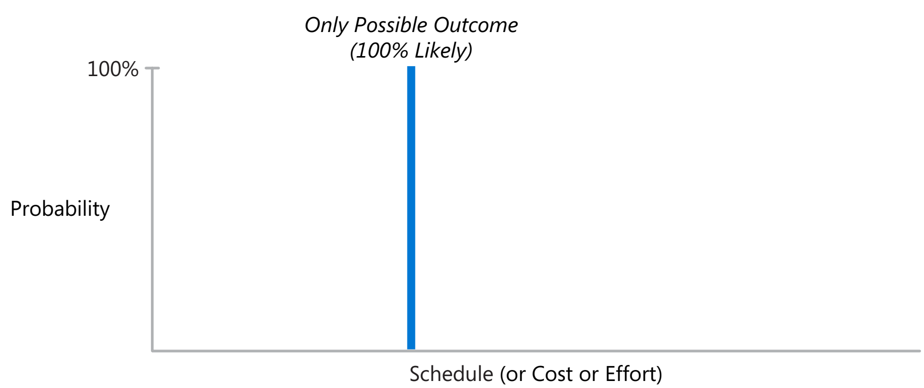


Figure 1-1 Single-point estimates assume 100% probability of the actual outcome equaling the planned outcome. This isn't realistic.

A single-point estimate is usually a target masquerading as an estimate. Occasionally, it is the sign of a more sophisticated estimate that has been stripped of meaningful probability information somewhere along the way.

Tip #3 When you see a single-point "estimate," ask whether the number is an estimate or whether it's really a target.

Accurate software estimates acknowledge that software projects are assailed by uncertainty from all quarters. Collectively, these various sources of uncertainty mean that project outcomes follow a probability distribution—some outcomes are more likely, some outcomes are less likely, and a cluster of outcomes in the middle of the distribution are most likely. You might expect that the distribution of project outcomes would look like a common bell curve, as shown in Figure 1-2.

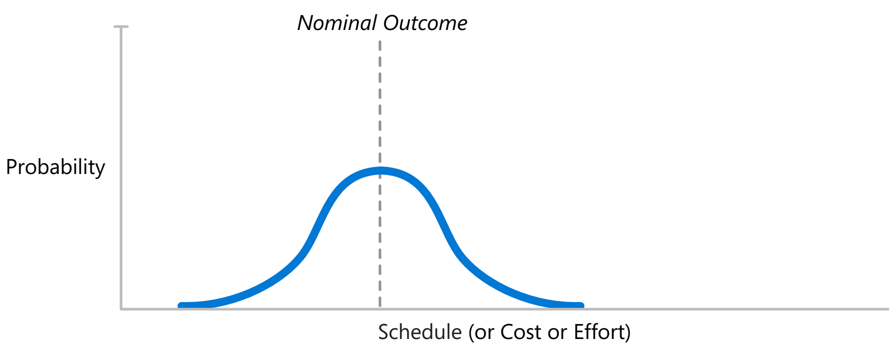


Figure 1-2 A common assumption is that software project outcomes follow a bell curve. This assumption is incorrect because there are limits to how efficiently a project team can complete any given amount of work.

Each point on the curve represents the chance of the project finishing exactly on that date (or costing exactly that much). The area under the curve adds up to 100%. This sort of probability distribution acknowledges the possibility of a broad range of outcomes. But the assumption that the outcomes are symmetrically distributed about the mean (average) is not valid. There is a limit to how well a project can be conducted, which means that the tail on the left side of the distribution is truncated rather than extending as far to the left as it does in the bell curve. And while there is a limit to how well a project can go, there is no limit to how poorly a project can go, and so the probability distribution does have a very long tail on the right.

Figure 1-3 provides an accurate representation of the probability distribution of a software project's outcomes.

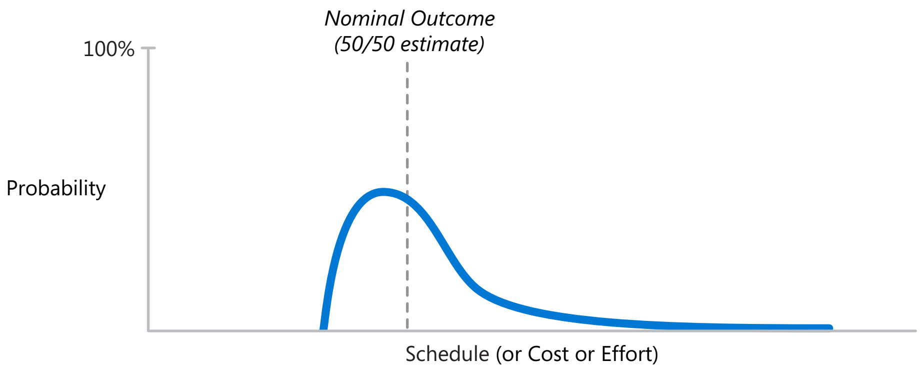


Figure 1-3 An accurate depiction of possible software project outcomes. There is a limit to how well a project can go but no limit to how many problems can occur.

The vertical dashed line shows the "nominal" outcome, which is also the "50/50" outcome—there's a 50% chance that the project will finish better and a 50% chance that it will finish worse. Statistically, this is known as the "median" outcome.

Figure 1-4 shows another way of expressing this probability distribution. While Figure 1-3 showed the probabilities of delivering on specific dates, Figure 1-5 shows the probabilities of delivering on each specific date *or earlier*.

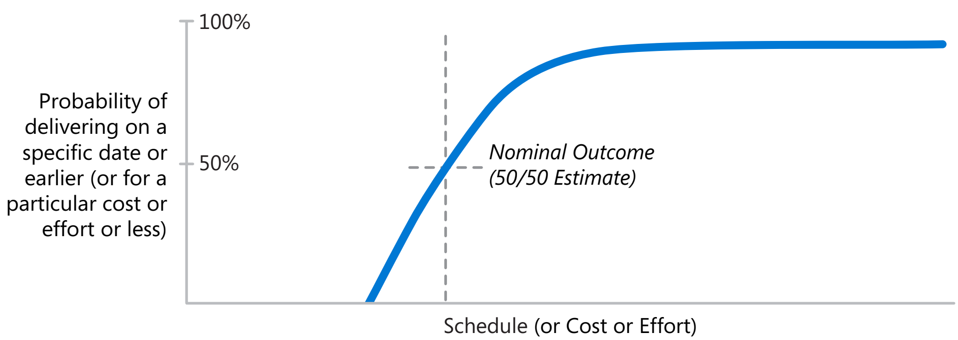


Figure 1-4 The probability of a software project delivering on or before a particular date (or less than or equal to a specific cost or level of effort).

Figure 1-5 presents the idea of probabilistic project outcomes in another way. As you can see from the figure, a naked estimate like "18 weeks" leaves out the interesting information that 18 weeks is only 10% likely. An estimate like "18 to 24 weeks" is more informative and conveys useful information about the likely range of project outcomes.

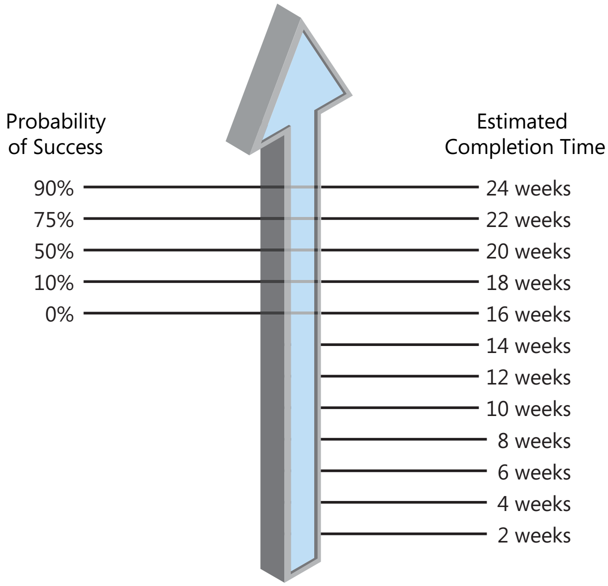


Figure 1-5 All single-point estimates are associated with a probability, explicitly or implicitly.

> **Tip #4** When you see a single-point estimate, that number's probability is not 100%. Ask what the probability of that number is.

You can express probabilities associated with estimates in numerous ways. You could use a "percent confident" attached to a single-point number: "We're 90% confident in the 24-week schedule." You could describe estimates as best case and worst case, which implies a probability: "We estimate a best case of 18 weeks and a worst case of 24 weeks." Or you could simply state the estimated outcome as a range rather than a single-point number: "We're estimating 18 to 24 weeks." The key point is that all estimates include a probability, whether the probability is stated or implied. An explicitly stated probability is one sign of a good estimate.

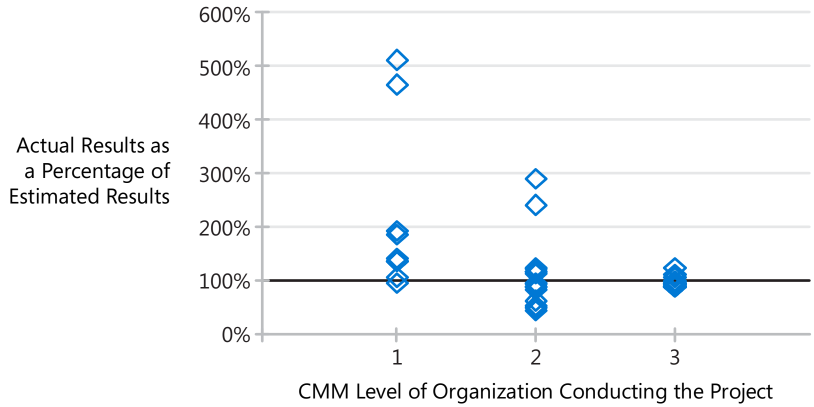
You can make a commitment to the optimistic end or the pessimistic end of an estimation range—or anywhere in the middle. The important thing is for you to know where in the range your commitment falls so that you can plan accordingly.

1.5 Common Definitions of a "Good" Estimate

The answer to the question of what an "estimate" is still leaves us with the question of what a *good* estimate is. Estimation experts have proposed various definitions of a good estimate. Capers Jones has stated that accuracy with ±10% is possible, but only on well-controlled projects (Jones 1998). Chaotic projects have too much variability to achieve that level of accuracy.

In 1986, Professors S.D. Conte, H.E. Dunsmore, and V.Y. Shen proposed that a good estimation approach should provide estimates that are within 25% of the actual results 75% of the time (Conte, Dunsmore, and Shen 1986). This evaluation standard is the most common standard used to evaluate estimation accuracy (Stutzke 2005).

Numerous companies have reported estimation results that are close to the accuracy Conte, Dunsmore, and Shen and Jones have suggested. Figure 1-6 shows actual results compared to estimates from a set of U.S. Air Force projects.



Source: "A Correlational Study of the CMM and Software Development Performance" (Lawlis, Flowe, and Thordahl 1995).

Figure 1-6 Improvement in estimation of a set of U.S. Air Force projects. The predictability of the projects improved dramatically as the organizations moved toward higher CMM levels.[1]

Figure 1-7 shows results of a similar improvement program at the Boeing Company.

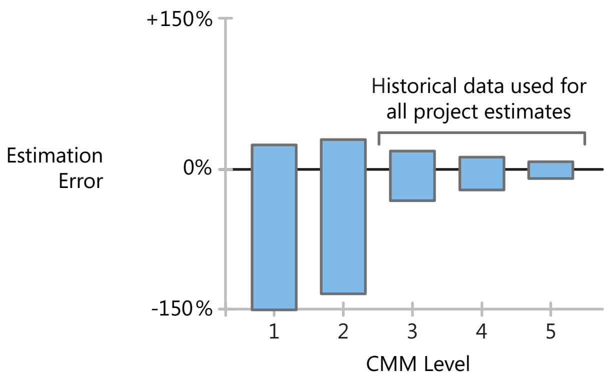


Figure 1-7 Improvement in estimation at the Boeing Company. As with the U.S. Air Force projects, the predictability of the projects improved dramatically at higher CMM levels.

[1] The CMM (Capability Maturity Model) is a system defined by the Software Engineering Institute to assess the effectiveness of software organizations.

A final, similar example, shown in Figure 1-8, comes from improved estimation results at Schlumberger.

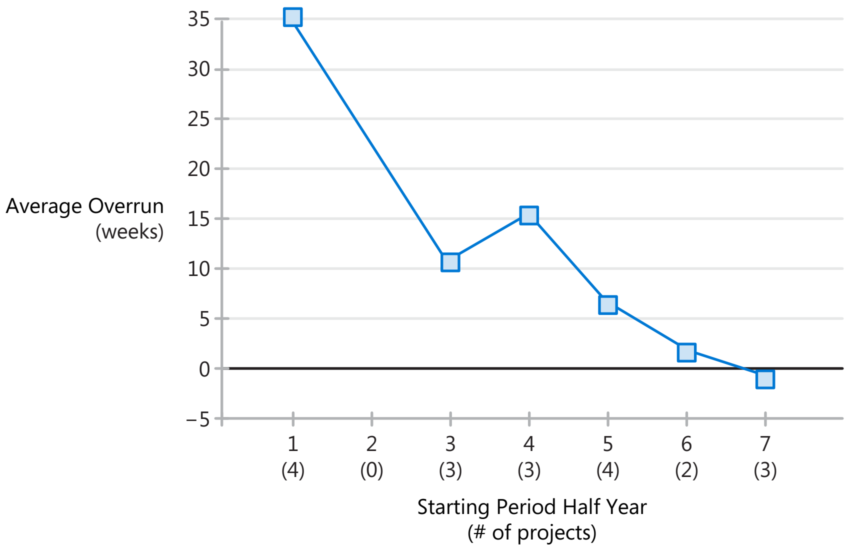


Figure 1-8 Schlumberger improved its estimation accuracy from an average overrun of 35 weeks to an average underrun of 1 week.

One of my client companies delivers 97% of its projects on time and within budget. Telcordia reported that it delivers 98% of its projects on time and within budget (Pitterman 2000). Numerous other companies have published similar results (Putnam and Myers 2003). Organizations are creating good estimates by both Jones's definition and Conte, Dunsmore, and Shen's definition. However, an important concept is missing from both of these definitions—namely, that accurate estimation results cannot be accomplished through estimation practices alone. They must also be supported by effective project control.

1.6 Estimates and Project Control

Sometimes when people discuss software estimation they treat estimation as a purely predictive activity. They act as though the estimate is made by an impartial estimator, sitting somewhere in outer space, disconnected from project planning and prioritization activities.

In reality, there is little that is pure about software estimation. If you ever wanted an example of Heisenberg's Uncertainty Principle applied to software, estimation would be it. (Heisenberg's Uncertainty Principle is the idea that the mere act of observing a thing changes it, so you can never be sure how that thing would behave if you weren't observing it.) Once we make an estimate and, on the basis of that estimate, make a commitment to deliver functionality and quality by a particular date, then we *control* the project to meet the target. Typical project control activities include removing non-critical requirements, redefining requirements, replacing less-experienced staff with more-experienced staff, and so on. Figure 1-9 illustrates these dynamics.

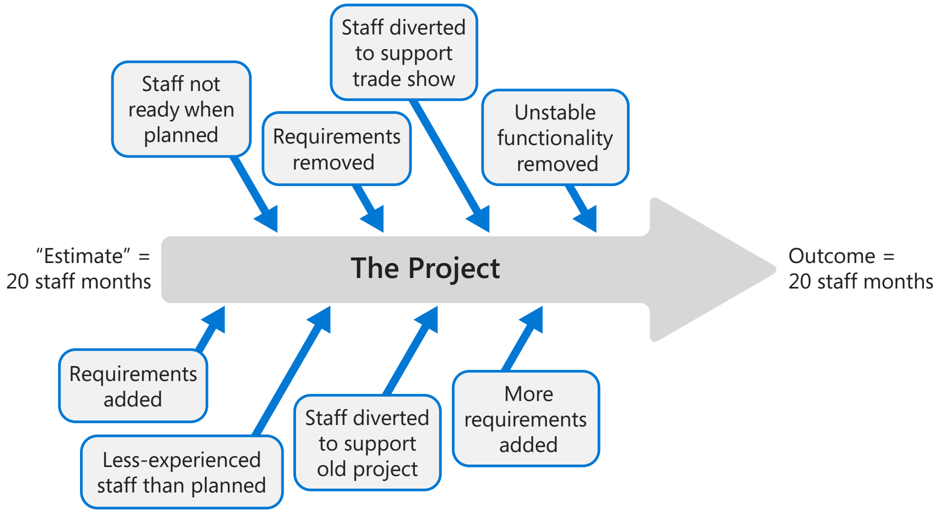


Figure 1-9 Projects change significantly from inception to delivery. Changes are usually significant enough that the project delivered is not the same as the project that was estimated. Nonetheless, if the outcome is similar to the estimate, we say the project met its estimate.

In addition to project control activities, projects are often affected by unforeseen external events. The project team might need to create an interim release to support a key customer. Staff might be diverted to support an old project, and so on.

Events that happen during the project nearly always invalidate the assumptions that were used to estimate the project in the first place. Functionality assumptions change, staffing assumptions change, and priorities change. It becomes impossible to make a clean analytical assessment of whether the project was estimated accurately—because the software project that was ultimately delivered is not the project that was originally estimated.

In practice, if we deliver a project with about the level of functionality intended, using about the level of resources planned, in about the time frame targeted, then we typically say that the project "met its estimates," despite all the analytical impurities implicit in that statement.

Thus, the criteria for a "good" estimate cannot be based on its predictive capability, which is impossible to assess, but on the estimate's ability to support project success, which brings us to the next topic: the Proper Role of Estimation.

1.7 Estimation's Real Purpose

Suppose you're preparing for a trip and deciding which suitcase to take. You have a small suitcase that you like because it's easy to carry and will fit into an airplane's overhead storage bin. You also have a large suitcase, which you don't like because you'll have to check it in and then wait for it at baggage claim, lengthening your trip. You lay your clothes beside the small suitcase, and it appears that they will almost fit. What do you do? You might try packing them very carefully, not wasting any space, and hoping they all fit. If that approach doesn't work, you might try stuffing them into the suitcase with brute force, sitting on the top and trying to squeeze the latches closed. If that still doesn't work, you're faced with a choice: leave a few clothes at home or take the larger suitcase.

Software projects face a similar dilemma. Project planners often find a gap between a project's business targets and its estimated schedule and cost. If the gap is small, the planner might be able to control the project to a successful conclusion by preparing extra carefully or by squeezing the project's schedule, budget, or feature set. If the gap is large, the project's targets must be reconsidered.

The primary purpose of software estimation is not to predict a project's outcome; it is to determine whether a project's targets are realistic enough to allow the project to be controlled to meet them. Will the clothes you want to take on your trip fit into the small suitcase or will you be forced to take the large suitcase? Can you take the small suitcase if you make minor adjustments? Executives want the same kinds of answers. They often don't want an accurate estimate that tells them that the desired clothes won't fit into the suitcase; they want a plan for making as many of the clothes fit as possible.

Problems arise when the gap between the business targets and the schedule and effort needed to achieve those targets becomes too large. I have found that if the initial target and initial estimate are within about 20% of each other, the project manager will have enough maneuvering room to control the feature set, schedule, team size, and other parameters to meet the project's business goals; other experts concur (Boehm 1981, Stutzke 2005). If the gap between the target and what is actually needed is too large, the manager will not be able to control the project to a successful conclusion by making minor adjustments to project parameters. No amount of careful packing or sitting on the suitcase will squeeze all your clothes into the smaller suitcase, and you'll have to take the larger one, even if it isn't your first choice, or you'll have to leave some clothes behind. The project targets will need to be brought into better alignment with reality before the manager can control the project to meet its targets.

Estimates don't need to be perfectly accurate as much as they need to be *useful*. When we have the combination of accurate estimates, good target setting, and good planning and control, we can end up with project results that are close to the "estimates." (As you've guessed, the word "estimate" is in quotation marks because the project that was estimated is not the same project that was ultimately delivered.)

These dynamics of changing project assumptions are a major reason that this book focuses more on the art of estimation than on the science. Accuracy of ±5% won't do you much good if the project's underlying assumptions change by 100%.

1.8 A Working Definition of a "Good Estimate"

With the background provided in the past few sections, we're now ready to answer the question of what qualifies as a good estimate.

> *A good estimate is an estimate that provides a clear enough view of the project reality to allow the project leadership to make good decisions about how to control the project to hit its targets..*

This definition is the foundation of the estimation discussion throughout the rest of this book.

Additional Resources

Conte, S. D., H. E. Dunsmore, and V. Y. Shen. *Software Engineering Metrics and Models*. Menlo Park, CA: Benjamin/Cummings, 1986. Conte, Dunsmore, and Shen's book contains the definitive discussion of evaluating estimation models. It discusses the "within 25% of actual 75% of the time" criteria, as well as many other evaluation criteria.

DeMarco, Tom. *Controlling Software Projects*. New York, NY: Yourdon Press, 1982. DeMarco discusses the probabilistic nature of software projects.

Stutzke, Richard D. *Estimating Software-Intensive Systems*. Upper Saddle River, NJ: Addison-Wesley, 2005. Appendix C of Stutzke's book contains a summary of measures of estimation accuracy.

Index

Numeric

A

B

C

D

E

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Steve McConnell

Steve McConnell is Chief Software Engineer at Construx Software where he oversees Construx's software engineering practices. Steve is the lead for the Construction Knowledge Area of the Software Engineering Body of Knowledge (SWEBOK) project. Steve has worked on software projects at Microsoft, Boeing, and other Seattle-area companies. Steve was the lead developer of Construx Estimate and of SPC Estimate Professional, winner of a *Software Development* magazine Productivity Award.

Steve is the author of *Rapid Development* (1996), *Software Project Survival Guide* (1998), *Professional Software Development* (2004) and *Code Complete, Second Edition* (2004). His books have twice won *Software Development* magazine's Jolt Product Excellence Award for outstanding software development book of the year. In 1998, readers of *Software Development* magazine named Steve one of the three most influential people in the software industry, along with Bill Gates and Linus Torvalds.

Steve earned a Bachelor's degree from Whitman College and a Master's degree in software engineering from Seattle University. He lives in Bellevue, Washington.

If you have any comments or questions about this book, please contact Steve at *steve.mcconnell@construx.com* or via *www.stevemcconnell.com*.

Additional Resources for Developers: Advanced Topics and Best Practices

Published and Forthcoming Titles from Microsoft Press

Code Complete, Second Edition

Steve McConnell ● ISBN 0-7356-1967-0

For more than a decade, Steve McConnell, one of the premier authors and voices in the software community, has helped change the way developers write code—and produce better software. Now his classic book, *Code Complete*, has been fully updated and revised with best practices in the art and science of constructing software. Topics include design, applying good techniques to construction, eliminating errors, planning, managing construction activities, and relating personal character to superior software. This new edition features fully updated information on programming techniques, including the emergence of Web-style programming, and integrated coverage of object-oriented design. You'll also find new code examples—both good and bad—in C++, Microsoft® Visual Basic®, C#, and Java, although the focus is squarely on techniques and practices.

More About Software Requirements: Thorny Issues and Practical Advice

Karl E. Wiegers ● ISBN 0-7356-2267-1

Have you ever delivered software that satisfied all of the project specifications, but failed to meet any of the customers expectations? Without formal, verifiable requirements—and a system for managing them—the result is often a gap between what developers think they're supposed to build and what customers think they're going to get. Too often, lessons about software requirements engineering processes are formal or academic, and not of value to real-world, professional development teams. In this follow-up guide to *Software Requirements*, Second Edition, you will discover even more practical techniques for gathering and managing software requirements that help you deliver software that meets project and customer specifications. Succinct and immediately useful, this book is a must-have for developers and architects.

Software Estimation: Demystifying the Black Art

Steve McConnell ● ISBN 0-7356-0535-1

Often referred to as the "black art" because of its complexity and uncertainty, software estimation is not as hard or mysterious as people think. However, the art of how to create effective cost and schedule estimates has not been very well publicized. *Software Estimation* provides a proven set of procedures and heuristics that software developers, technical leads, and project managers can apply to their projects. Instead of arcane treatises and rigid modeling techniques, award-winning author Steve McConnell gives practical guidance to help organizations achieve basic estimation proficiency and lay the groundwork to continue improving project cost estimates. This book does not avoid the more complex mathematical estimation approaches, but the non-mathematical reader will find plenty of useful guidelines without getting bogged down in complex formulas.

Debugging, Tuning, and Testing Microsoft .NET 2.0 Applications

John Robbins ● ISBN 0-7356-2202-7

Making an application the best it can be has long been a time-consuming task best accomplished with specialized and costly tools. With Microsoft Visual Studio® 2005, developers have available a new range of built-in functionality that enables them to debug their code quickly and efficiently, tune it to optimum performance, and test applications to ensure compatibility and trouble-free operation. In this accessible and hands-on book, debugging expert John Robbins shows developers how to use the tools and functions in Visual Studio to their full advantage to ensure high-quality applications.

The Security Development Lifecycle

Michael Howard and Steve Lipner ● ISBN 0-7356-2214-0

Adapted from Microsoft's standard development process, the Security Development Lifecycle (SDL) is a methodology that helps reduce the number of security defects in code at every stage of the development process, from design to release. This book details each stage of the SDL methodology and discusses its implementation across a range of Microsoft software, including Microsoft Windows Server™ 2003, Microsoft SQL Server™ 2000 Service Pack 3, and Microsoft Exchange Server 2003 Service Pack 1, to help measurably improve security features. You get direct access to insights from Microsoft's security team and lessons that are applicable to software development processes worldwide, whether on a small-scale or a large-scale. This book includes a CD featuring videos of developer training classes.

Software Requirements, Second Edition
Karl E. Wiegers ● ISBN 0-7356-1879-8

Writing Secure Code, Second Edition
Michael Howard and David LeBlanc ● ISBN 0-7356-1722-8

CLR via C#, Second Edition
Jeffrey Richter ● ISBN 0-7356-2163-2

For more information about Microsoft Press® books and other learning products, visit: **www.microsoft.com/mspress** *and* **www.microsoft.com/learning**

Microsoft Press products are available worldwide wherever quality computer books are sold. For more information, contact your book or computer retailer, software reseller, or local Microsoft Sales Office, or visit our Web site at **www.microsoft.com/mspress**. To locate your nearest source for Microsoft Press products, or to order directly, call 1-800-MSPRESS in the United States. (In Canada, call **1-800-268-2222**.)

Additional Resources for Web Developers

Published and Forthcoming Titles from Microsoft Press

Microsoft® Visual Web Developer™ 2005 Express Edition: Build a Web Site Now!
Jim Buyens • ISBN 0-7356-2212-4

With this lively, eye-opening, and hands-on book, all you need is a computer and the desire to learn how to create Web pages now using Visual Web Developer Express Edition! Featuring a full working edition of the software, this fun and highly visual guide walks you through a complete Web page project from set-up to launch. You'll get an introduction to the Microsoft Visual Studio® environment and learn how to put the light-weight, easy-to-use tools in Visual Web Developer Express to work right away—building your first, dynamic Web pages with Microsoft ASP.NET 2.0. You'll get expert tips, coaching, and visual examples at each step of the way, along with pointers to additional learning resources.

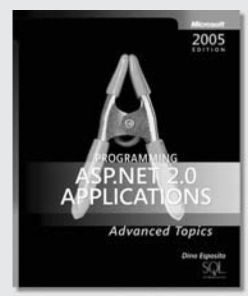
Microsoft ASP.NET 2.0 Programming *Step by Step*
George Shepherd • ISBN 0-7356-2201-9

With dramatic improvements in performance, productivity, and security features, Visual Studio 2005 and ASP.NET 2.0 deliver a simplified, high-performance, and powerful Web development experience. ASP.NET 2.0 features a new set of controls and infrastructure that simplify Web-based data access and include functionality that facilitates code reuse, visual consistency, and aesthetic appeal. Now you can teach yourself the essentials of working with ASP.NET 2.0 in the Visual Studio environment—one step at a time. With *Step by Step*, you work at your own pace through hands-on, learn-by-doing exercises. Whether you're a beginning programmer or new to this version of the technology, you'll understand the core capabilities and fundamental techniques for ASP.NET 2.0. Each chapter puts you to work, showing you how, when, and why to use specific features of the ASP.NET 2.0 rapid application development environment and guiding you as you create actual components and working applications for the Web, including advanced features such as personalization.

Programming Microsoft ASP.NET 2.0 *Core Reference*
Dino Esposito • ISBN 0-7356-2176-4

Delve into the core topics for ASP.NET 2.0 programming, mastering the essential skills and capabilities needed to build high-performance Web applications successfully. Well-known ASP.NET author Dino Esposito deftly builds your expertise with Web forms, Visual Studio, core controls, master pages, data access, data binding, state management, security services, and other must-know topics—combining definitive reference with practical, hands-on programming instruction. Packed with expert guidance and pragmatic examples, this *Core Reference* delivers the key resources that you need to develop professional-level Web programming skills.

Programming Microsoft ASP.NET 2.0 Applications: *Advanced Topics*
Dino Esposito • ISBN 0-7356-2177-2



Master advanced topics in ASP.NET 2.0 programming—gaining the essential insights and in-depth understanding that you need to build sophisticated, highly functional Web applications successfully. Topics include Web forms, Visual Studio 2005, core controls, master pages, data access, data binding, state management, and security considerations. Developers often discover that the more they use ASP.NET, the more they need to know. With expert guidance from ASP.NET authority Dino Esposito, you get the in-depth, comprehensive information that leads to full mastery of the technology.

Programming Microsoft Windows® Forms
Charles Petzold • ISBN 0-7356-2153-5

Programming Microsoft Web Forms
Douglas J. Reilly • ISBN 0-7356-2179-9

CLR via C++
Jeffrey Richter with Stanley B. Lippman
ISBN 0-7356-2248-5

Debugging, Tuning, and Testing Microsoft .NET 2.0 Applications
John Robbins • ISBN 0-7356-2202-7

CLR via C#, Second Edition
Jeffrey Richter • ISBN 0-7356-2163-2

For more information about Microsoft Press® books and other learning products, visit: **www.microsoft.com/books** *and* **www.microsoft.com/learning**

Microsoft® Press

Microsoft Press products are available worldwide wherever quality computer books are sold. For more information, contact your book or computer retailer, software reseller, or local Microsoft Sales Office, or visit our Web site at **www.microsoft.com/mspress**. To locate your nearest source for Microsoft Press products, or to order directly, call 1-800-MSPRESS in the United States. (In Canada, call **1-800-268-2222**.)

Additional SQL Server Resources for Developers

Published and Forthcoming Titles from Microsoft Press

Microsoft® SQL Server™ 2005 Express Edition
Step by Step

Jackie Goldstein ● ISBN 0-7356-2184-5

Teach yourself how to get database projects up and running quickly with SQL Server Express Edition—a free, easy-to-use database product that is based on SQL Server 2005 technology. It's designed for building simple, dynamic applications, with all the rich functionality of the SQL Server database engine and using the same data access APIs, such as Microsoft ADO.NET, SQL Native Client, and T-SQL. Whether you're new to database programming or new to SQL Server, you'll learn how, when, and why to use specific features of this simple but powerful database development environment. Each chapter puts you to work, building your knowledge of core capabilities and guiding you as you create actual components and working applications.

Microsoft SQL Server 2005 Programming
Step by Step

Fernando Guerrero ● ISBN 0-7356-2207-8

SQL Server 2005 is Microsoft's next-generation data management and analysis solution that delivers enhanced scalability, availability, and security features to enterprise data and analytical applications while making them easier to create, deploy, and manage. Now you can teach yourself how to design, build, test, deploy, and maintain SQL Server databases—one step at a time. Instead of merely focusing on describing new features, this book shows new database programmers and administrators how to use specific features within typical business scenarios. Each chapter provides a highly practical learning experience that demonstrates how to build database solutions to solve common business problems.

Microsoft SQL Server 2005 Analysis Services
Step by Step

Hitachi Consulting Services ● ISBN 0-7356-2199-3

One of the key features of SQL Server 2005 is SQL Server Analysis Services—Microsoft's customizable analysis solution for business data modeling and interpretation. Just compare SQL Server Analysis Services to its competition to understand the great value of its enhanced features. One of the keys to harnessing the full functionality of SQL Server will be leveraging Analysis Services for the powerful tool that it is—including creating a cube, and deploying, customizing, and extending the basic calculations. This step-by-step tutorial discusses how to get started, how to build scalable analytical applications, and how to use and administer advanced features. Interactivity (enhanced in SQL Server 2005), data translation, and security are also covered in detail.

Microsoft SQL Server 2005 Reporting Services
Step by Step

Hitachi Consulting Services ● ISBN 0-7356-2250-7

SQL Server Reporting Services (SRS) is Microsoft's customizable reporting solution for business data analysis. It is one of the key value features of SQL Server 2005: functionality more advanced and much less expensive than its competition. SRS is powerful, so an understanding of how to architect a report, as well as how to install and program SRS, is key to harnessing the full functionality of SQL Server. This procedural tutorial shows how to use the Report Project Wizard, how to think about and access data, and how to build queries. It also walks through the creation of charts and visual layouts for maximum visual understanding of data analysis. Interactivity (enhanced in SQL Server 2005) and security are also covered in detail.

Programming Microsoft SQL Server 2005

Andrew J. Brust, Stephen Forte, and William H. Zack
ISBN 0-7356-1923-9

This thorough, hands-on reference for developers and database administrators teaches the basics of programming custom applications with SQL Server 2005. You will learn the fundamentals of creating database applications—including coverage of T-SQL, Microsoft .NET Framework, and Microsoft ADO.NET. In addition to practical guidance on database architecture and design, application development, and reporting and data analysis, this essential reference guide covers performance, tuning, and availability of SQL Server 2005.

Inside Microsoft SQL Server 2005:
The Storage Engine
Kalen Delaney ● ISBN 0-7356-2105-5

Inside Microsoft SQL Server 2005:
T-SQL Programming
Itzik Ben-Gan ● ISBN 0-7356-2197-7

Inside Microsoft SQL Server 2005:
Query Processing and Optimization
Kalen Delaney ● ISBN 0-7356-2196-9

Programming Microsoft ADO.NET 2.0 Core Reference
David Sceppa ● ISBN 0-7356-2206-X

For more information about Microsoft Press® books and other learning products, visit: **www.microsoft.com/mspress** *and* **www.microsoft.com/learning**

Microsoft Press

Microsoft Press products are available worldwide wherever quality computer books are sold. For more information, contact your book or computer retailer, software reseller, or local Microsoft Sales Office, or visit our Web site at **www.microsoft.com/mspress**. To locate your nearest source for Microsoft Press products, or to order directly, call 1-800-MSPRESS in the United States. (In Canada, call **1-800-268-2222**.)

Additional Resources for Visual Basic Developers

Published and Forthcoming Titles from Microsoft Press

Microsoft® Visual Basic® 2005 Express Edition: Build a Program Now!

Patrice Pelland • ISBN 0-7356-2213-2

Featuring a full working edition of the software, this fun and highly visual guide walks you through a complete programming project—a desktop weather-reporting application—from start to finish. You'll get an introduction to the Microsoft Visual Studio® development environment and learn how to put the lightweight, easy-to-use tools in Visual Basic Express to work right away—creating, compiling, testing, and delivering your first ready-to-use program. You'll get expert tips, coaching, and visual examples each step of the way, along with pointers to additional learning resources.

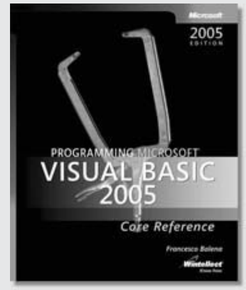
Microsoft Visual Basic 2005 *Step by Step*

Michael Halvorson • ISBN 0-7356-2131-4

With enhancements across its visual designers, code editor, language, and debugger that help accelerate the development and deployment of robust, elegant applications across the Web, a business group, or an enterprise, Visual Basic 2005 focuses on enabling developers to rapidly build applications. Now you can teach yourself the essentials of working with Visual Studio 2005 and the new features of the Visual Basic language—one step at a time. Each chapter puts you to work, showing you how, when, and why to use specific features of Visual Basic and guiding as you create actual components and working applications for Microsoft Windows®. You'll also explore data management and Web-based development topics.

Programming Microsoft Visual Basic 2005 *Core Reference*

Francesco Balena • ISBN 0-7356-2183-7



Get the expert insights, indispensable reference, and practical instruction needed to exploit the core language features and capabilities in Visual Basic 2005. Well-known Visual Basic programming author Francesco Balena expertly guides you through the fundamentals, including modules, keywords, and inheritance, and builds your mastery of more advanced topics such as delegates, assemblies, and My Namespace. Combining in-depth reference with extensive, hands-on code examples and best-practices advice, this *Core Reference* delivers the key resources that you need to develop professional-level programming skills for smart clients and the Web.

Programming Microsoft Visual Basic 2005 Framework Reference

Francesco Balena • ISBN 0-7356-2175-6

Complementing *Programming Microsoft Visual Basic 2005 Core Reference*, this book covers a wide range of additional topics and information critical to Visual Basic developers, including Windows Forms, working with Microsoft ADO.NET 2.0 and ASP.NET 2.0, Web services, security, remoting, and much more. Packed with sample code and real-world examples, this book will help developers move from understanding to mastery.

Programming Microsoft Windows Forms
Charles Petzold • ISBN 0-7356-2153-5

Programming Microsoft Web Forms
Douglas J. Reilly • ISBN 0-7356-2179-9

Debugging, Tuning, and Testing Microsoft .NET 2.0 Applications
John Robbins • ISBN 0-7356-2202-7

Microsoft ASP.NET 2.0 *Step by Step*
George Shepherd • ISBN 0-7356-2201-9

Microsoft ADO.NET 2.0 *Step by Step*
Rebecca Riordan • ISBN 0-7356-2164-0

Programming Microsoft ASP.NET 2.0 *Core Reference*
Dino Esposito • ISBN 0-7356-2176-4

For more information about Microsoft Press® books and other learning products, visit: **www.microsoft.com/books** *and* **www.microsoft.com/learning**

Microsoft® Press

Additional SQL Server Resources for Administrators

Published and Forthcoming Titles from Microsoft Press

Microsoft® SQL Server™ 2005 Reporting Services *Step by Step*

Hitachi Consulting Services • ISBN 0-7356-2250-7

SQL Server Reporting Services (SRS) is Microsoft's customizable reporting solution for business data analysis. It is one of the key value features of SQL Server 2005: functionality more advanced and much less expensive than its competition. SRS is powerful, so an understanding of how to architect a report, as well as how to install and program SRS, is key to harnessing the full functionality of SQL Server. This procedural tutorial shows how to use the Report Project Wizard, how to think about and access data, and how to build queries. It also walks the reader through the creation of charts and visual layouts to enable maximum visual understanding of the data analysis. Interactivity (enhanced in SQL Server 2005) and security are also covered in detail.

Microsoft SQL Server 2005 Administrator's Pocket Consultant

William R. Stanek • ISBN 0-7356-2107-1

Here's the utterly practical, pocket-sized reference for IT professionals who need to administer, optimize, and maintain SQL Server 2005 in their organizations. This unique guide provides essential details for using SQL Server 2005 to help protect and manage your company's data—whether automating tasks; creating indexes and views; performing backups and recovery; replicating transactions; tuning performance; managing server activity; importing and exporting data; or performing other key tasks. Featuring quick-reference tables, lists, and step-by-step instructions, this handy, one-stop guide provides fast, accurate answers on the spot, whether you're at your desk or in the field!

Microsoft SQL Server 2005 Administrator's Companion

Marci Frohock Garcia, Edward Whalen, and Mitchell Schroeter • ISBN 0-7356-2198-5

Microsoft SQL Server 2005 Administrator's Companion is the comprehensive, in-depth guide that saves time by providing all the technical information you need to deploy, administer, optimize, and support SQL Server 2005. Using a hands-on, example-rich approach, this authoritative, one-volume reference book provides expert advice, product information, detailed solutions, procedures, and real-world troubleshooting tips from experienced SQL Server 2005 professionals. This expert guide shows you how to design high-availability database systems, prepare for installation, install and configure SQL Server 2005, administer services and features, and maintain and troubleshoot your database system. It covers how to configure your system for your I/O system and model and optimize system capacity. The expert authors provide details on how to create and use defaults, constraints, rules, indexes, views, functions, stored procedures, and triggers. This guide shows you how to administer reporting services, analysis services, notification services, and integration services. It also provides a wealth of information on replication and the specifics of snapshot, transactional, and merge replication. Finally, there is expansive coverage of how to manage and tune your SQL Server system, including automating tasks, backup and restoration of databases, and management of users and security.

Microsoft SQL Server 2005 Analysis Services *Step by Step*

Hitachi Consulting Services • ISBN 0-7356-2199-3

One of the key features of SQL Server 2005 is SQL Server Analysis Services—Microsoft's customizable analysis solution for business data modeling and interpretation. Just compare SQL Server Analysis Services to its competition to understand/grasp the great value of its enhanced features. One of the keys to harnessing the full functionality of SQL Server will be leveraging Analysis Services for the powerful tool that it is—including creating a cube, and deploying, customizing, and extending the basic calculations. This step-by-step tutorial discusses how to get started, how to build scalable analytical applications, and how to use and administer advanced features. Interactivity (which is enhanced in SQL Server 2005), data translation, and security are also covered in detail.

Microsoft SQL Server 2005 Express Edition
Step by Step
Jackie Goldstein • ISBN 0-7356-2184-5

Inside Microsoft SQL Server 2005:
The Storage Engine
Kalen Delaney • ISBN 0-7356-2105-5

Inside Microsoft SQL Server 2005:
T-SQL Programming
Itzik Ben-Gan • ISBN 0-7356-2197-7

Inside Microsoft SQL Server 2005:
Query Processing and Optimization
Kalen Delaney • ISBN 0-7356-2196-9

For more information about Microsoft Press® books and other learning products, visit: **www.microsoft.com/mspress** *and* **www.microsoft.com/learning**

Microsoft Press

Microsoft Press products are available worldwide wherever quality computer books are sold. For more information, contact your book or computer retailer, software reseller, or local Microsoft Sales Office, or visit our Web site at **www.microsoft.com/mspress**. To locate your nearest source for Microsoft Press products, or to order directly, call 1-800-MSPRESS in the United States. (In Canada, call **1-800-268-2222**.)

Prepare for Certification with Self-Paced Training Kits

Official Exam Prep Guides—
Plus Practice Tests

Ace your preparation for the skills measured by the MCP exams—and on the job. With official *Self-Paced Training Kits* from Microsoft, you'll work at your own pace through a system of lessons, hands-on exercises, troubleshooting labs, and review questions. Then test yourself with the Readiness Review Suite on CD, which provides hundreds of challenging questions for in-depth self-assessment and practice.

- **MCSE Self-Paced Training Kit (Exams 70-290, 70-291, 70-293, 70-294): Microsoft® Windows Server™ 2003 Core Requirements.** 4-Volume Boxed Set. ISBN: 0-7356-1953-0. (Individual volumes are available separately.)
- **MCSA/MCSE Self-Paced Training Kit (Exam 70-270): Installing, Configuring, and Administering Microsoft Windows® XP Professional, Second Edition.** ISBN: 0-7356-2152-7.
- **MCSE Self-Paced Training Kit (Exam 70-298): Designing Security for a Microsoft Windows Server 2003 Network.** ISBN: 0-7356-1969-7.
- **MCSA/MCSE Self-Paced Training Kit (Exam 70-350): Implementing Microsoft Internet Security and Acceleration Server 2004.** ISBN: 0-7356-2169-1.
- **MCSA/MCSE Self-Paced Training Kit (Exam 70-284): Implementing and Managing Microsoft Exchange Server 2003.** ISBN: 0-7356-1899-2.

For more information about Microsoft Press® books, visit: **www.microsoft.com/mspress**

For more information about learning tools such as online assessments, e-learning, and certification, visit: **www.microsoft.com/mspress** *and* **www.microsoft.com/learning**

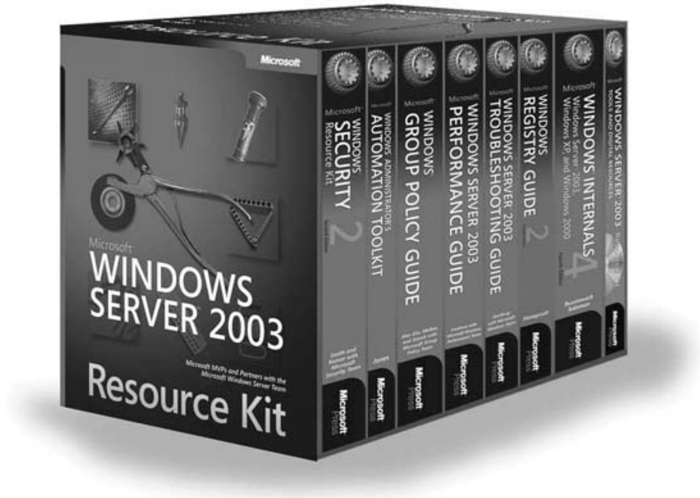
Microsoft® Press

Microsoft Press products are available worldwide wherever quality computer books are sold. For more information, contact your book or computer retailer, software reseller, or local Microsoft Sales Office, or visit our Web site at **www.microsoft.com/mspress**. To locate your nearest source for Microsoft Press products, or to order directly, call 1-800-MSPRESS in the United States. (In Canada, call **1-800-268-2222**.)

Microsoft Windows Server 2003 Resource Kit

The *definitive* resource

for Windows Server 2003!



Get the in-depth technical information and tools you need to manage and optimize Microsoft® Windows Server™ 2003—with expert guidance and best practices from Microsoft MVPs, leading industry consultants, and the Microsoft Windows Server team. This official *Resource Kit* delivers seven comprehensive volumes, including:

- **Microsoft Windows® Security Resource Kit, Second Edition**
- **Microsoft Windows Administrator's Automation Toolkit**
- **Microsoft Windows Group Policy Guide**
- **Microsoft Windows Server 2003 Performance Guide**
- **Microsoft Windows Server 2003 Troubleshooting Guide**
- **Microsoft Windows Registry Guide, Second Edition**
- **Microsoft Windows Internals, Fourth Edition**

You'll find 300+ timesaving tools and scripts, an eBook of the entire *Resource Kit*, plus five bonus eBooks. It's everything you need to help maximize system performance and reliability—and help reduce ownership and support costs.

Microsoft Windows Server 2003 Resource Kit
Microsoft MVPs and Partners with the Microsoft Windows Server Team
ISBN: 0-7356-2232-9

For more information about Microsoft Press® books, visit: **www.microsoft.com/mspress**

For more information about learning tools such as online assessments, e-learning, and certification, visit: **www.microsoft.com/learning**

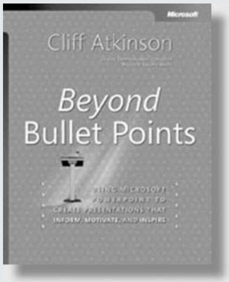
Microsoft Press products are available worldwide wherever quality computer books are sold. For more information, contact your book or computer retailer, software reseller, or local Microsoft Sales Office, or visit our Web site at **www.microsoft.com/mspress**. To locate your nearest source for Microsoft Press products, or to order directly, call 1-800-MSPRESS in the United States. (In Canada, call **1-800-268-2222**.)

Additional Resources for Business and Home Users

Published and Forthcoming Titles from Microsoft Press

Beyond Bullet Points: Using Microsoft® PowerPoint® to Create Presentations That Inform, Motivate, and Inspire

Clift Atkinson • ISBN 0-7356-2052-0



Improve your presentations—and increase your impact—with 50 powerful, practical, and easy-to-apply techniques for Microsoft PowerPoint. With *Beyond Bullet Points*, you'll take your presentation skills to the next level—learning innovative ways to design and deliver your message. Organized into five sections, including Distill Your Ideas, Structure Your Story, Visualize Your Message, Create a Conversation, and Maintain Engagement—the book uses clear, concise language and just the right visuals to help you understand concepts and start getting better results.

Take Back Your Life! Special Edition: Using Microsoft Outlook® to Get Organized and Stay Organized

Sally McGhee • ISBN 0-7356-2215-9

Unrelenting e-mail. Conflicting commitments. Endless interruptions. In this book, productivity expert Sally McGhee shows you how to take control and reclaim something that you thought you'd lost forever—your work-life balance. Now you can benefit from Sally's popular and highly regarded corporate education programs, learning simple but powerful techniques for rebalancing your personal and professional commitments using the productivity features in Outlook. When you change your approach, you can change your results. So learn what thousands of Sally's clients worldwide have discovered about taking control of their everyday productivity—and start transforming your own life today!

On Time! On Track! On Target! Managing Your Projects Successfully with Microsoft Project

Bonnie Biafore • ISBN 0-7356-2256-6

This book focuses on the core skills you need to successfully manage any project, giving you a practical education in project management and how-to instruction for using Microsoft Office Project Professional 2003 and other Microsoft Office Professional Edition 2003 programs, such as Excel® 2003, Outlook 2003, and Word 2003. Learn the essentials of project management, including creating successful project plans, tracking and evaluating performance, and controlling project costs. Whether you're a beginner just learning how to manage projects or a project manager already working on a project, this book has something for you. Includes a companion CD with sample Project templates.

Design to Sell: Using Microsoft Publisher to Inform, Motivate, and Persuade

Roger C. Parker • ISBN 0-7356-2260-4

Design to Sell relates the basics of effective message creation and formatting to the specific capabilities built into Microsoft Publisher—the powerful page layout program found on hundreds of thousands of computers around the world. Many Microsoft Office users already have Publisher on their computers but don't use it because they don't think of themselves as writers or designers. Here is a one-stop guide to marketing that even those without big budgets or previous design or writing experience can use to create compelling, easy-to-read marketing materials. Each chapter has an interactive exercise as well as questions with answers on the author's Web site. Also on the Web site are downloadable worksheets and templates, book updates, more illustrations of the projects in the book, and additional before-and-after project makeovers.

Microsoft Windows® XP Networking and Security Inside Out: Also Covers Windows 2000

Ed Bott and Carl Siechert • ISBN 0-7356-2042-3

Configure and manage your PC network—and help combat privacy and security threats—from the inside out! Written by the authors of the immensely popular *Microsoft Windows XP Inside Out*, this book packs hundreds of timesaving solutions, troubleshooting tips, and work-arounds for networking and security topics—all in concise, fast-answer format.

Dig into the tools and techniques for configuring workgroup, domain, Internet, and remote networking, and all the network components and features in between. Get the answers you need to use Windows XP Service Pack 2 and other tools, tactics, and features to help defend your personal computer and network against spyware, pop-up ads, viruses, hackers, spam, denial-of-service attacks, and other threats. Learn how to help secure your Virtual Private Networks (VPNs), remote access, and wireless networking services, and take ultimate control with advanced solutions such as file encryption, port blocking, IPSec, group policies, and tamper-proofing tactics for the registry. Get up to date on hot topics such as peer-to-peer networks, public wireless access points, smart cards, handheld computers, wireless LANs, and more. Plus, the CD includes bonus resources that make it easy for you to share your new security and networking expertise with your colleagues, friends, and family.