DHQ

THE DORSET HOUSE QUARTERLY VOL. XIII NO. 2 FALL 2003

DORSET HOUSE PUBLISHING 353 WEST 12TH STREET NEW YORK, NEW YORK 10014 USA

ABOUT THE BOOK

with

Lawrence H. Putnam

and **Ware Myers**

authors of

Five Core Metrics: The Intelligence Behind Successful Software Management

ISBN: 0-932633-55-2 ©2003
328 pages softcover $49.95 ppd.

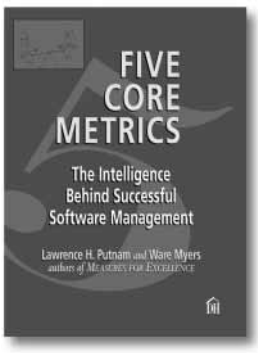
DHQ: *What are the "five core metrics" featured in your new book? When in the development process do you take those measurements?*

LHP: The five core metrics are size, schedule time, effort (cost), defects, and process productivity. You use them from the start of and then throughout the life of the development process, whenever necessary and appropriate.

WM: In the first two phases of the Unified Process, Inception and Elaboration, one of the purposes is to figure out sufficiently what is to be done, in order to attach a size estimate to the project. So, thinking about what is to be done (especially in terms of size) goes all the way back to the beginning.

DHQ: *Who on the development team is responsible for recording metrics? Does it require additional staff?*

LHP: The project manager is responsible overall. He or she usually has a quality assurance team as part of the development team. This is often a very good place to put the metrics collection and analysis function. It can be done as a part-time effort. It usually does not require a full-time dedicated person. Tools are useful for recording and processing the metrics data. The tools can be homegrown using spreadsheets. Also, there are excellent commercial tools that have been specially developed for this purpose. Such tools aid in collecting the metrics data and also provide analytical capabilities to interpret the data and tell the story the metrics are designed to convey.



DHQ: *How often do you record the metrics?*

LHP: Monthly or weekly during development, so you have a good history of what went on during the project.

DHQ: *How often and when do you do analysis work with the metrics?*

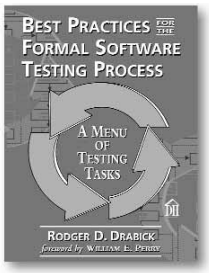
LHP: Do analysis at the start of the project, using historic data and estimates of the key parameters for the new project. This is related to the estimate for the new project. Update this estimate whenever the scope or other key factors of the project change. Use the monthly or weekly metrics data to do comparison analysis as soon as the project is under way. By the time the project is

(continued on page 2)

Greetings

With this issue of *DHQ*, we are pleased to announce that two new books are in stock and selling briskly: *Best Practices for the Formal Software Testing Process: A Menu of Testing Tasks*, by Rodger D. Drabick, and *Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency*, by Tom DeMarco. See below and page 7.

A Soup-to-Nuts Process for Formal System Testing



Best Practices for the Formal Software Testing Process, by Rodger D. Drabick, is now in stock. Drabick shows testers how to develop a formal testing process that best suits their needs and those of their organization and its customers. See page 4 for an excerpt.

Put Your Slack to Good Use with Tom DeMarco

Tom DeMarco's *Slack* is now available for the first time from Dorset House Publishing. Originally published by Broadway Books in 2001, this hardcover title debunks the myth of total efficiency, showing how efficiency improvement efforts can backfire—slowing things down rather than fostering agility, responsiveness, and speed. DeMarco's characteristically counterintuitive solution is for companies to build slack into their processes. Here, slack puts effectiveness ahead of absolute efficiency, giving workers room to breathe and innovate. Slack isn't a luxury—it's a competitive strategy. David A. Kaplan, author of *Silicon Boys*, writes, "This book is the ideal tonic to the '90s craze of downsizing, restructuring, cost-cutting—all in the name of efficiency and global competition." Read more on page 3.

Praise for *Waltzing with Bears*

As Richard Mateosian writes in his *IEEE Micro* review of DeMarco and Lister's new book, "If you have anything to do with software, you really need the information in this book." He also writes that "the people at Dorset House Publishing have found a great formula and I hope they stick to it." Thank you, Richard!

New Catalog in Print and PDF

The 2004 Catalog of Books is here. Request a printed copy or download the PDF version on Dorsethouse.com.

Forthcoming Books Galore

See page 6 for a hearty crop of titles under development. You'll receive a 20 percent discount when you order early—see page 7.

Order Online with PayPal

Dorsethouse.com now offers secure online purchasing through free PayPal accounts. The PayPal shopping cart allows you to order by credit card using an existing PayPal account or one set up on-the-fly. We're ready for your order!

INSIDE *DHQ*

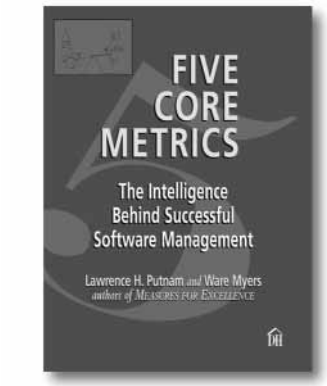
Is this your first *DHQ*?

Send your postal address for a free subscription, but also send your e-mail address for **e-DHQ**, our monthly e-mail newsletter. Since we only mail ***DHQ*** to subsets of our list, **e-DHQ** is the most reliable way to get ***DHQ***'s news and features. We maintain our list in-house, and we do not rent, sell, or share information about our customers with any third-parties.

Call 1-800-342-6657 or (212) 620-4053, fax (212) 727-1044, e-mail info@dorsethouse.com, or visit us at www.dorsethouse.com/comments.html.

Editor: David McClintock
Editorial Asssts.: Nuno Andrade and Vincent Au

NOW IN STOCK



Five Core Metrics:

The Intelligence Behind Successful Software Management

by Lawrence H. Putnam *and* Ware Myers

ISBN: 0-932633-55-2 ©2003
328 pages softcover $49.95 postpaid

To succeed in the software industry, software developers need to cultivate a reliable development process. By measuring what teams have achieved on previous projects, developers can more accurately set goals, make bids, and ensure the successful completion of new projects.

Acclaimed long-time collaborators Lawrence H. Putnam and Ware Myers present simple but powerful measurement techniques to help software developers allocate limited resources and track project progress.

Drawing new findings from an extensive database of software project metrics, the authors demonstrate how readers can control projects with just five core metrics—Time, Effort, Size, Reliability, and Process Productivity. With these metrics, developers can adjust ongoing projects to changing conditions—surprises that would otherwise cause project failure.

ABOUT THE AUTHORS

Larry Putnam, Sr., and Ware Myers have written three previous books and numerous articles together over many years. Mr. Putnam, a leading expert in the software estimation and management field, is the president of Quantitative Software Management, a software management consulting firm based in McLean, Virginia.

Ware Myers is an independent consultant and a long-time contributing editor to *IEEE Computer* and *IEEE Software*. His current interests include the application of metrics to software planning, estimating, bidding, and project control.

Read More About This Book at www.dorsethouse.com/books/fcm.html

About the Book *(continued from page 1)*

roughly 25 percent complete in schedule, you can compare actuals against the plan and make an adaptive update forecast if actuals are deviating significantly from plan. At the end of the project, capture all the key information relating to the size, schedule, effort, and defects. Use this to determine the process productivity achieved. Store this in your metrics repository to use on new projects coming along.

WM: One addition that I would make to what Larry said is something about the psychology of data collection. Management should make it clear that this data is being collected for metric purposes, not for personnel use. People are willing to collect data honestly for estimating and control purposes, but if they suspect it is to have something to do with personnel appraising, they begin to fudge the data and consequently to distort the data for metric purposes.

DHQ: *What's new about relating metrics to the Unified Process, and why isn't it enough to follow the process without metrics?*

LHP: Metrics can be a part of any development methodology and should be. The originators of the Unified Process did not make metrics a part of their original creation. However, the company they are affiliated with, Rational, has been integrating metrics with a number of its products and is beginning to infuse its thinking with metrics concepts. Metrics ideas are important to the management of development because they allow resource requirements for development to be integrated into the planning and control process that is at the heart of management. This permits you to put the right number of staff on the project at the right time; it permits the prediction of when defects will occur and how many, so defect elimination is timely; it allows you to project ahead to determine when the defect discovery rate will be low enough that the product can be released for use by the end user. These metrics functions are important to the planning and control of any development process. There is a significant void in development when metrics are not used.

DHQ: *Won't keeping track of metrics make it more difficult to follow a formal software development process?*

LHP: Keeping track of metrics during development is not an onerous task. It can be handled part-time by project support people (usually in quality assurance). The volume of data collected is very modest when focused on the five core metrics we discuss in our book. There are good commercial tools available that collect data, help with analysis, and plan and control the development process, so the time and effort spent servicing the metrics work is very modest and not costly.

WM: It is perfectly true that collecting metric data takes a little time and making use of it takes a little more time. But it is customary on a planet of limited resources—limited daytime, limited lifespans, and so on—to try to get work done within some limits. In fact, we have a name for it: competition. And those who don't come close to competition-set limits are removed from business (and even government). In software development, the name for this set of activities is metrics. To the extent that software development is an economic activity, not a weekend pastime, it has to base itself on metrics, and most organizations do use some kind of metrics. The issue, really, is whether the metrics are good or "seat of the pants." Part of making them good is the realization that they are tied into process, models, and tools.

DHQ: *Don't metrics bog-down the speed and creativity of software development?*

LHP: No. Actually, they speed it up because there are fewer false starts and fewer rework cycles. They can be handled as a by-product of the development process. Moreover, they provide valuable insights into what is happening on the project while it is under way, and they afford an opportunity to make timely changes to the project by providing near-real-time observation of problems that may ultimately cause cost and schedule overruns.

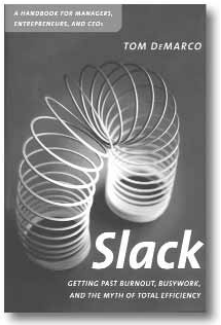
DHQ: *Thank you!*

There's More to This Interview
Visit **www.dorsethouse.com/features/interviews/intfcm.html#more**.

DHQ Excerpt

Efficiency and/or Effectiveness

by **Tom DeMarco**



Adapted from Chapter 17 of

Slack:
Getting Past Burnout, Busywork, and the Myth of Total Efficiency

ISBN: 0-932633-61-7 ©2001
240 pages hardcover $25.95 postpaid

The overstressed organization is so busy making itself efficient that it has clean forgotten how to be effective. The two are not at all the same. You're efficient when you do something with minimum waste. And you're effective when you're doing the right something. It's possible to be one without the other: efficient but not effective, or effective but not efficient.

Of course, it is also possible to be both. Possible, but not easy. You ought not to be obliged to choose strictly between the two, but suppose you were. Which one would you choose? Efficiency or effectiveness? That's pretty easy. An effective but not efficient organization moves steadily (though maybe not quickly) toward its real goals. How much progress it makes in that direction is a matter of how inefficient it is. An efficient but not effective organization, on the other hand, is moving in the wrong direction. The more it optimizes, the more progress it makes away from its real goals. Such an organization could say of itself, in Yogi Berra's words, "We're lost, but we're making good time."

Why Achieving Both Is Not Easy

Let's face it, the implicit goal in all organizations is to be both: to make effective choices about what to do and then carry those choices out efficiently. That presumption is so strongly built into organizational cultures everywhere that their executives sometimes can't see when it isn't happening. It's absolutely supposed to be happening, so it must be. The fact that the organization *is* moving in a given direction is strong *a priori* evidence that it must be the *right* direction. Executives are annoyed when anyone in the organization challenges direction. "We wouldn't be doing this stuff at all if it weren't right; now what we need is for everyone to get on board to help us do it as efficiently as possible."

Unfortunately, momentum in some direction or other does not necessarily imply carefully thought-out strategic thinking. A company can begin to move (or be moved) by a process that is more or less *drift*. The Brownian motion within the company asserts a net force in some direction and "By God we're moving." The difference between strategic thinking and drift is a matter of whether the key choices are made mindfully or mindlessly.

It may sound like a harsh charge that organizations are setting directions mindlessly, that they're prone to get their tactics right but not their strategy. But tactics are a lot easier than strategy. Tactics can be handled in isolation. You as head of a single department in your company can optimize that department to make it more efficient in what it does, but you can't unilaterally redirect it to do something different. That change would have to be effected above you, where the issues are an order of magnitude more complex. And it would have to be done in such a way as to build wide consensus among disparate interest groups. This requires both powerful vision and charismatic leadership. The idea that drift is often substituted for strategic direction-setting is no more surprising than the observation that visionary, charismatic leaders are few and far between.

All this suggests that a lot of companies are not really led at all. If that's true, why isn't it more apparent? Why don't they seem leaderless? That is the direct result of what I call the Easy Executive Option:

> *Directing an entire organization is hard. Seeming to direct it, on the other hand, is easy. All you have to do is note which way the drift is moving and instruct the organization to go that way.*

It was the Easy Executive Option at work, for example, that caused General Motors to cede the small-vehicle sector to foreign competition and to lag behind during the Eighties and Nineties in energy-efficient engines and nontraditional fuels.

In addition to being flat-out hard to do, building effectiveness into an organization often comes into direct conflict with increasing efficiency. This is an unfortunate side effect of optimization, first noted by the geneticist R.A. Fisher, and now referred to as Fisher's fundamental theorem: "The more highly adapted an organism becomes, the less adaptable it is to any new change." Fisher's example was the giraffe. It is highly adapted to food found up among the tree branches, but so unadaptable to a new situation that it can not even pick a up a peanut from the ground at the zoo.

The more efficient your organization has become, the more it's going to need the steps laid out in the next few chapters to improve effectiveness. Taking those steps is not going to be trivial, but the alternative is to proceed more and more efficiently away from your real goals.

"Reading this book clears up the trade-offs between efficiency and effectiveness, between doing and planning, between switching and concentration, and how squeezing excess capacity out of your company can sometimes leave it terminally unresponsive." —**Bob Metcalfe**
inventor of the Ethernet, founder of 3Com
author of *Internet Collapses*

About the Author

Tom DeMarco is a principal of the Atlantic Systems Guild and author or coauthor of four best-selling Dorset House books (*Peopleware, Software State-of-the-Art, Waltzing with Bears,* and *Why Does Software Cost So Much?)* and a groundbreaking training video (*Productive Teams,* with Timothy Lister).

DHQ Excerpt

Why We Need the Formal Testing Process Model

by **Rodger D. Drabick**

Adapted from Chapter 1 of

Best Practices for the Formal Software Testing Process

ISBN: 0-932633-58-7 ©2004
312 pages softcover $41.95 postpaid

As the testing life cycle has matured and test techniques have developed, new test engineers and test managers have to acquire additional knowledge of the test domain. Since testing is more than simply a phase of the software development life cycle, people performing the test engineering and test manager functions must be aware of this fact. We must become salespeople of a sort, to transfer this knowledge to our executive management. If we don't have this knowledge ourselves, we won't be very good at the sales part of the job.

But, if you as a new test engineer or a new test manager haven't spent years in the field, how do you know what to sell? As an example, the key to developing a good test program is a good test plan, and the key to developing a good test plan is a good set of requirements. There is a variety of literature that discusses techniques for generating good requirements, but there are a lot of very talented software development engineers and test engineers who have never read any of that literature. The situation relative to executive management is even worse. We test engineers should be stressing the need for good requirements development to the development and management communities. We can do this in three ways: by encouraging requirements reviews, by reviewing requirements testability using a simple technique, and by developing test plans.

How do you know that this is something you should be doing? You could spend a year attending testing conferences sponsored by a variety of organizations; two of the best are the Quality Assurance Institute's International Conferences on Software Testing and Software Quality Engineering's STAR Conferences. Or, you could read the IEEE Software Engineering Standards from cover to cover (note that, currently, this is a four-volume set). Alternatively, if there were one concise roadmap of the tasks a mature test engineering organization can and should be performing, it would assist in identifying these tasks. I submit to you, that if you become acquainted with the process model described in this book, you have this roadmap in your hands.

In brief, what I am providing in this book is a list of tasks and processes for a program of formal testing. If you are just starting to implement a testing process, you will not want to try to implement all parts of this model in one fell swoop; it would be too overwhelming and too costly. In later chapters, I suggest ways to implement this process in a prioritized, piecewise fashion.

Cooperation Between Testers and Developers

It is worth noting that as test engineering has matured since the early 1970's, we have realized that we do our best work in an environment where there is no longer an adversarial relationship between the people doing software development and those folks performing the role of test engineering. We are all part of one organization, and our overriding goal is to produce the best possible systems and software for our customers, given the constraints of resources and time.

One of the reasons for reviewing requirements and addressing testability of those requirements early in the life cycle is to make testing easier, but testable requirements will also make design and coding easier for software development. If requirements are so ambiguous that test engineers don't know what to test, how can developers be expected to design and code to those requirements?

In contrast to some of my contemporaries, I have always tried to encourage complete and open communication between a test team and a development team. Software development personnel should be reviewing test documentation, in the same way that test engineers review requirements, design, and code. Early and frequent feedback between the various groups on a program will contribute to lower costs and faster time-to-market. There should be no surprises in the test documentation for development staff members, since they should have thoroughly reviewed these products.

How to Use the Formal Testing Process Model

There is no "one true way" to test software, so the goal of this book is to provide you with a structure and set of best practices that you can use to develop a formal testing process that fits your needs, and those of your customers, your management, and your organization. You should consider this book as a menu of things you could do, and then establish a set of priorities for what you'll do first, what you'll do second, and so on.

As an example, suppose you are working in a test engineering role in a company in which the current practice is that when executive management gets an idea for a new application to add to the software suite your company produces, the following happens: First, management assigns a developer to the task; then, the developer takes the limited input from management and writes code for the application. Thus, in such an environment, the entire requirements and design process is minimized at best, and short-circuited at worst. When code is complete, as far as the developer is concerned (and we'll believe that he or she did some limited unit testing based on an understanding of

(continued on page 5)

About the Author

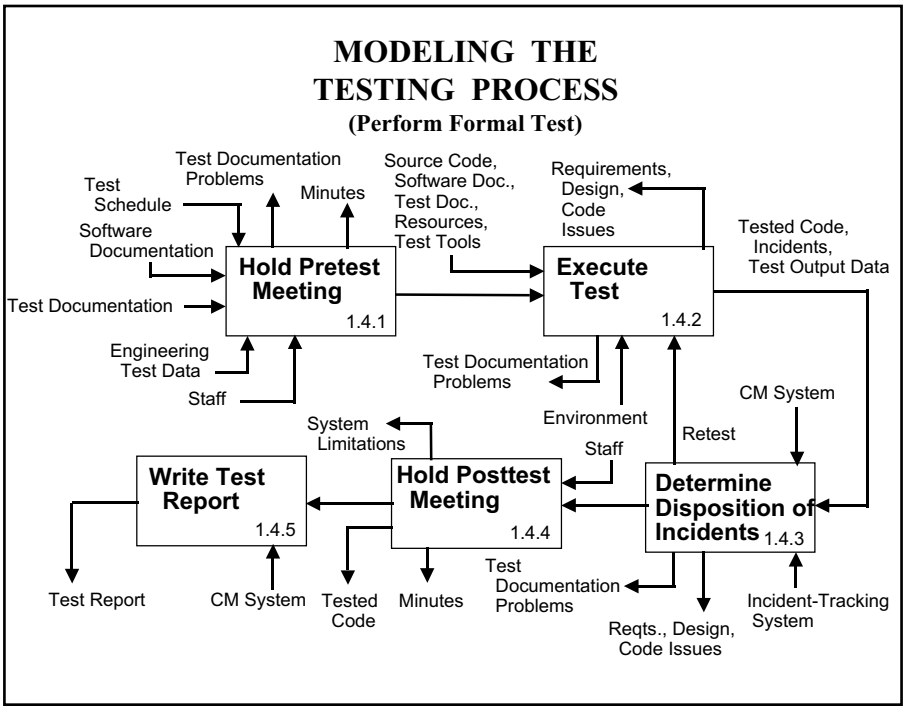
Rodger D. Drabick is a nationally recognized quality engineering and systems testing expert with extensive experience in the Capability Maturity Model for software and process improvement. With nearly three decades in software quality assurance and testing, he has been responsible for SQA initiatives and developing process improvement action plans at companies such as Amtrak, Bell-Atlantic, the Federal Aviation Authority, Kodak, and Lockheed Martin.

The Need for the Formal Testing Process Model *(continued from page 4)*

the "requirements"), the developer hands the code to the test engineer he or she is most comfortable working with, or to the team leader of testing. This may well be the first time that the folks involved in testing have heard about this new or upgraded application.

If you're in the role of testing team leader or test engineer, what good does the model in this book do you? In this specific instance, I'd advise you to look carefully at the Perform Formal Test process (see figure below). Consider which of the five subprocesses (Hold Pretest Meeting, Execute Test, Determine Disposition of Incidents, Hold Posttest Meeting, or Write Test Report) you are currently performing. I suspect that, in the environment I've postulated, your group is only performing the Execute Test and Determine Disposition of Incidents subprocesses. It would not be difficult to start holding pretest and posttest meetings, and these could markedly improve your process. At the pretest meeting, you should ask the developer to describe the unit and integration testing performed and to identify areas of risk. The personnel assigned to testing can then concentrate on these risk areas. In such an environment, there probably hasn't been time to develop a test plan or test design, but what would you base the plan or design on, anyway? Test cases or test procedures will have been developed based on discussions with the developer and the development team lead or manager. These can also be reviewed at the pretest meeting for adequacy. This could turn out to be a long meeting for a complex application. You might want to schedule a test-case and test-procedure review meeting prior to the pretest meeting. Peer reviews of test documentation are always a good idea, and the earlier they can be held, the better the return.

As part of the Perform Formal Test process, you should also make sure you have some sort of tool or process in place to record defects discovered in testing, a process to present them to the developers, a process to incorporate the fixes, and a process to regression-test the application to make sure that the fixes work (and, perhaps equally important, that they don't break something else). ■



Perform Formal Test: The steps engineers must take to prepare for testing, execute the test steps, handle incidents, determine success or failure of the test, and write the formal test report.

Recent Reviews

Five Core Metrics
by **Lawrence H. Putnam** and **Ware Myers**
ISBN: 0-932633-55-2 ©2003 328 pages
softcover $49.95 (includes $6 for UPS in US)

". . . an invaluable resource for project managers and project leaders. Those managers bombarded with software measurements will find what Putnam and Myers refer to as the 'five core metrics' to be a welcome solution to their metric woes."
—**Michelle Giles**, *StickyMinds.com*

". . . a must-read for managers who want to bring development under control.

"The book is intended to be used by software development managers, and their bosses, and provides a comprehensive approach to achieving predictability in the software development process." —**Joe Saur**
ACM Software Engineering Notes

Waltzing with Bears
by **Tom DeMarco** and **Timothy Lister**
ISBN: 0-932633-60-9 ©2003 208 pages
softcover $33.95 (includes $6 for UPS in US)

"The authors invest a good deal of energy into fighting with general practice. Statements like 'Risk management is project management for adults' are meant to needle people into changing the status quo.

"If you have anything to do with software, you really need the information in this book." —**Richard Mateosian**
IEEE Micro

"I liked this book. It caused me to reexamine the way I budget software as well as other deeper assumptions about what I 'choose to believe' or rather why I choose to believe it. If you get a chance to read the book, you will probably come to the same conclusion, 'Wow, I never thought about things that way, and I really think I should!'" —**Will Tracz**
ACM Software Engineering Notes

Book Proposal Guidelines

Manuscripts are welcome on topics in systems and software development. We seek unique, usable insights on analysis, design, testing, and management, ideas that serve to inspire and improve the efforts of professional IS developers and managers.

Send us your bio, a topic discussion, an assessment of competing books, a detailed table of contents, and as many sample chapters as are available. Let us know your ideas for promoting the book. Text-only, Word, PDF, or other files are welcome by e-mail to info@dorsethouse.com

Please note that due to the rigor and selectivity of our review process—and the busy schedules of our expert reviewers—our response time varies from six to ten weeks, sometimes longer. Simultaneous submissions are welcome, but please notify us of competing offers.

Book Proposals
Dorset House Publishing
353 West 12th Street
New York, NY 10014

ORDER EARLY & SAVE*

Agile Software Development in the Large: Diving Into the Deep
by JUTTA ECKSTEIN
ISBN: 0-932633-57-9 ©2004 est. 216 pages softcover $39.95 ppd.

Most agile processes have been developed to support small to mid-sized software development projects. This book shows how to apply the principles and values of agile processes to large teams. Topics include • the effect of the principles and value system of agile processes on large teams • the impact on a team of a switch to an agile process • the characteristics of the process that will allow you to coordinate several sub-teams • how the size of the project and of the team influence the underlying architecture • and more.

Hiring Technical People: The Artful Science of Getting the Right Person for the Job
by JOHANNA ROTHMAN
ISBN: 0-932633-59-5 ©2004 est. 280 pages softcover $33.95 ppd.

Hiring technical people is one of the most critical and difficult processes a manager can undertake. This book takes the guesswork out of hiring, and diminishes the risk of costly hiring mistakes. With the aid of step-by-step descriptions and detailed examples, you'll learn how to • write a job description • source candidates • develop ads • review résumés • develop interview techniques • create phone-screens • check references • extend an offer • and more.

System Testing with an Attitude
by NATHAN PETSCHENIK
ISBN: 0-932633-46-3 ©2004 est. 306 pages softcover $45.95 ppd.

Developers striving for high quality and rapid time-to-market need to adopt an attitude: It is unacceptable for software that does not meet requirements to even reach the system test phase. Instituting this attitude requires not only the involvement of the system testers, but also that of the development team. *System Testing with an Attitude* explains how to cultivate productive relationships between developers and system testers and stresses the importance of identifying and delineating the responsibilities of each group, which can prevent problems in the system before system testing even begins.

Systems Modeling & Requirements Specification Using ECSAM: Embedded Computer-Based Systems Analysis
by JONAH Z. LAVI and JOSEPH KUDISH
ISBN: 0-932633-45-5 ©2004 est. 440 pages softcover $53.95 ppd.

Discover ECSAM, a method for requirements engineering and the modeling of computer-based systems (CBS). Practiced since 1980 in evolving versions by large numbers of systems and software engineers worldwide, ECSAM was developed in part at Israel Aircraft Industries for the analysis and design of complex reactive embedded systems and software. The method guides engineers in modeling operational, functional, and design requirements, considering both static and dynamic aspects of systems.

Testing Dirty Systems
by WILLIAM E. PERRY and RANDALL W. RICE
ISBN: 0-932633-56-0 ©2004 est. 368 pages softcover $41.95 ppd.

Some systems are more difficult to test than others. Software testers contend with undefined or partially defined requirements; outdated, incomplete, or nonexistent documentation; complex logic; a mixture of languages; or worse. All of these factors make a system dirty, or virtually untestable. In *Testing Dirty Systems,* William Perry and Randall Rice—authors of *Surviving the Top Ten Challenges of Software Testing*—teach testers a six-step process for approaching such systems: system diagnosis • test planning • test execution • test analysis • report development • dirty system repair. Project leaders, independent testers, quality assurance personnel, and IS auditors will benefit from this book, as well as end-users and customers with a vested interest in the success of their systems.

Save 20% On These and Other Titles—See Page 7 for Details.

Order Form

Call 1-800-342-6657 or (212) 620-4053. *Mention "DHQ V13N2."*
Fax to (212) 727-1044 *or* **Mail** *(see address below)*
Questions? Call or fax, **e-mail** info@dorsethouse.com, or visit **www.dorsethouse.com**.

✓***SAVE 20% ON UNDERLINED TITLES—CALL FOR DETAILS***

TITLE	#	PRICE	TOTAL
Adaptive Software Development (HIGHSMITH)		$44.95	
Agile Software Development in the Large (ECKSTEIN) ***FORTHCOMING***		$27.16†	
Amplifying Your Effectiveness (WEINBERG, BACH & KARTEN)		$24.95	
Are Your Lights On? (GAUSE & WEINBERG)		$13.95	
Becoming a Technical Leader (WEINBERG)		$29.95	
Best Practices for the Formal Software Testing Process (DRABICK) ***NEW***		$35.95	
Communication Gaps and How to Close Them (KARTEN)		$33.95	
Complete Systems Analysis (J. & S. ROBERTSON)		$57.95	
Creating a Software Engineering Culture (WIEGERS)		$39.95	
Data Model Patterns (HAY)		$39.95	
The Deadline: A Novel About Project Management (DeMARCO)		$24.95	
Designing Quality Databases with IDEF1X Information Models (BRUCE)		$57.95	
Dr. Peeling's Principles of Management (PEELING) ***NEW***		$29.95	
Everyday Heroes of the Quality Movement (GLUCKMAN & ROOME)		$19.95✓	
Exploring Requirements (GAUSE & WEINBERG)		$44.95	
Five Core Metrics (PUTNAM & MYERS) ***NEW***		$43.95	
Fundamentals of Object-Oriented Design in UML (PAGE-JONES)		$39.95*	
General Principles of Systems Design (D. & G. WEINBERG)		$27.95	
Handbook of Walkthroughs, Inspections, and Technical Rev. (FREEDMAN & WEINBERG)		$49.95	
Hiring Technical People (ROTHMAN) ***FORTHCOMING***		$22.36†	
How to Plan, Develop & Use Information Systems (VAN STEENIS)		$34.95✓	
An Intro. to General Systems Thinking: Silver Anniv. Edition (WEINBERG)		$33.95	
Managing Expectations (KARTEN)		$27.95	
Measuring and Managing Performance in Organizations (AUSTIN)		$24.95	
More Secrets of Consulting: The Consultant's Tool Kit (WEINBERG)		$33.95	
The One Minute Methodology (ORR)		$12.95✓	
Peopleware, 2nd ed. (DeMARCO & LISTER)		$33.95	
Practical Guide to Business Process Reengineering Using IDEF0 (FELDMANN)		$34.95	
Practical Project Management (PAGE-JONES)		$34.95✓	
Process for System Arch. and Requirements Eng. (HATLEY, HRUSCHKA & PIRBHAI)		$59.95	
Productive Teams: A Video (DeMARCO & LISTER)		$95.00✓	
Project Retrospectives (KERTH)		$33.95	
The Psychology of Computer Programming: Silver Anniversary Edition (WEINBERG)		$44.95	
Quality Software Management, Vol. 1: Systems Thinking (WEINBERG)		$41.95	
Quality Software Management, Vol. 2: First-Order Measurement (WEINBERG)		$43.95	
Quality Software Management, Vol. 3: Congruent Action (WEINBERG)		$39.95	
Quality Software Management, Vol. 4: Anticipating Change (WEINBERG)		$44.95	
Rethinking Systems Analysis & Design (WEINBERG)		$27.95	
Roundtable on Project Management (BULLOCK, WEINBERG & BENESH)		$15.95	
Roundtable on Technical Leadership (WEINBERG, BENESH & BULLOCK)		$15.95	
To Satisfy & Delight Your Customer (PARDEE)		$39.95✓	
The Secrets of Consulting (WEINBERG)		$29.95	
Slack (DeMARCO) ***NEW***		$19.95	
Software Productivity (MILLS)		$25.95✓	
Software State-of-the-Art (DeMARCO & LISTER)		$45.95✓	
Strategies for Real-Time System Specification (HATLEY & PIRBHAI)		$49.95	
Surviving the Top Ten Challenges of Software Testing (PERRY & RICE)		$27.95	
System Testing with an Attitude (PETSCHENIK) ***FORTHCOMING***		$28.76†	
Systems Modeling & Req. Spec. Using ECSAM (LAVI & KUDISH) ***FORTHCOMING***		$38.36†	
Testing Dirty Systems (PERRY & RICE) ***FORTHCOMING***		$28.76†	
Understanding the Professional Programmer (WEINBERG)		$24.95	
Waltzing with Bears (DeMARCO & LISTER) ***NEW***		$27.95	
What Every Programmer Should Know About Object-Oriented Design (PAGE-JONES)		$44.95▲	
Why Does Software Cost So Much? (DeMARCO)		$29.95	

*No qty. discount. ▲Qty. limited. See Dorsethouse.com for offer.
✓Call for details: Save 20% when you order with a full-priced title.
†Discounted price valid on orders received before the indicated expiration date (see left column).

SHIPPING AND HANDLING POLICY: First book or video, add $6.00 for UPS shipping (first class, air, and all shipments outside continental USA are additional: please call, fax, or e-mail with quantities desired, shipping address, and your contact information (phone, fax, e-mail); we will provide shipping costs estimate). Each additional book, volume, or video up to 5, add $1.25 per item. For 6 or more items, call us for actual charges. Please allow 1 to 4 weeks for delivery. We ship UPS, unless otherwise requested. For UPS, please give complete street address, no postal boxes. PAYMENT MUST ACCOMPANY ORDER in US funds; NYS residents, please add sales tax. Prices are effective Jan. 1, 2002 and are subject to change without notice.

SUBTOTAL:
SHIPPING:
NYS TAX:
TOTAL:

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

❑ VISA ❑ MC ❑ AMEX Card # _____

Exp. date _____ Signature _____

Daytime phone _____ e-mail address _____

❑ *E-mail me a shipping confirmation.* ❑ *Subscribe me to* **e-DHQ**.

NAME TITLE

COMPANY

❑ HOME ❑ BUSINESS STREET (NO P.O. BOXES FOR UPS DELIVERY)

CITY STATE ZIP *Callers, please mention* **DHQ V13N2**

DORSET HOUSE PUBLISHING 353 WEST 12TH STREET NEW YORK, NEW YORK 10014 USA www.dorsethouse.com

Order Early... Save 20%

Discount valid on prepaid orders received before the indicated expiration date.

Agile Software Development in the Large

by Jutta Eckstein
ISBN: 0-932633-57-9 ©2004
est. 216 pp. softcover
Reg.: $39.95 postpaid

Reserve Your Copy Before January 15, 2004 and Pay Only $33.16 postpaid

Hiring Technical People

by Johanna Rothman
ISBN: 0-932633-59-5 ©2004
est. 280 pp. softcover
Reg.: $33.95 postpaid

Reserve Your Copy Before January 15, 2004 and Pay Only $28.36 postpaid

System Testing with an Attitude

by Nathan Petschenik
ISBN: 0-932633-46-3 ©2004
est. 306 pp. softcover
Reg.: $45.95 postpaid

Reserve Your Copy Before February 15 2004 and Pay Only $37.96 postpaid

Systems Modeling & Requirements Specification Using ECSAM

by Jonah Z. Lavi and Joseph Kudish
ISBN: 0-932633-45-5 ©2004
est. 440 pp. softcover
Reg.: $53.95 postpaid

Reserve Your Copy Before January 15, 2004 and Pay Only $44.36 postpaid

Testing Dirty Systems

by William E. Perry and Randall W. Rice
ISBN: 0-932633-56-0 ©2004
est. 368 pp. softcover
Reg.: $41.95 postpaid

Reserve Your Copy Before February 15, 2004 and Pay Only $34.76 postpaid

Your Credit Card Won't Be Charged Until We're Ready to Process Your Shipment.

Call 1-800-342-6657
or (212) 620-4053
Weekdays 9AM-6PM EST.
VISA MC AMEX

Order Online

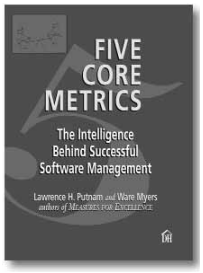
Now U.S. residents can use the PayPal shopping cart to order by credit card from **Dorsethouse.com**. PayPal accounts are free, and you can set one up when you order.

DHQ

THE DORSET HOUSE QUARTERLY VOL. XIII NO. 2 FALL 2003

DORSET HOUSE PUBLISHING 353 WEST 12TH STREET NEW YORK, NEW YORK 10014 USA

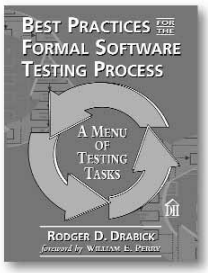
See Page 7 to Save 20% On Selected Titles



Inside, read an interview with **Lawrence H. Putnam** and **Ware Myers,** authors of **Five Core Metrics: The Intelligence Behind Successful Software Management**.

ISBN: 0-932633-55-2 ©2003
328 pages $33.95 postpaid

See page 1.



Inside, read an excerpt from Rodger D. Drabick's **Best Practices for the Formal Software Testing Process**.

ISBN: 0-932633-58-7 ©2004
312 pages $41.95 postpaid

See page 4.

Greetings

This issue of *The Dorset House Quarterly* offers an interview, excerpts, news about forthcoming books, reviews, and Author News.

If this is your first issue of *DHQ,* or if you'd like to receive *e-DHQ,* the e-mail edition, call us at 1-800-342-6657 to sign up for a free subscription.

Or, send us your address and phone number via our Website, **www.dorsethouse.com**, or e-mail **info@dorsethouse.com**.

We welcome your comments, requests, and questions.

Toll Free!
1-800-DH-BOOKS

THE DORSET HOUSE QUARTERLY
353 West 12th Street
New York, NY 10014

(212) 620-4053 FAX (212) 727-1044
www.dorsethouse.com info@dorsethouse.com

Please mail, fax, or e-mail address corrections/removals
ADDRESS SERVICE REQUESTED