

## Best Practices . . .

“. . . offers two major benefits to the reader. The first is when to test at points during the development life cycle, and the second is how to test at those points in the development life cycle.

“. . . contains literally hundreds of ideas. . . . If you only picked one good idea from the book, it would be more than worth the price.”

—William E. Perry

Executive Director, Quality Assurance Institute

“. . . a useful book for those working on test efforts in formal environments. . . . There have been plenty of templates and standards floating around for years on what to write down for such tests, but precious little describing how to manage the formal testing process. This book fills that void. . . .

“If you are testing in a formal environment for the first time, reading Rodger’s book might well go from a good idea to a survival requirement.”

—Rex Black

posted on Amazon.com

“This book is a fabulous primer for those faced with moving from an ad hoc or exploratory testing situation to a fully documented CMMI type process.

“This is especially valuable when you have an industrial giant, such as Boeing, looking over your shoulder and wondering what your standard processes are like. Using this book as a reference, I was able to put the right process in place.”

—David Tardiff

TYBRIN Corp.

“. . . a comprehensive and practical guide to formal software testing process. Everyone involved in software testing will benefit from his years of experience and his revealing insights. . . . a great textbook for new testers, a step-by-step cookbook for new managers, and a great reference book for everyone in the testing world.”

—Lisa Crispin

posted on Amazon.com

## 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.

## Partial Contents

### Preface

- 1: Introduction
- 2: The Formal Testing Process Model: Level 1 IPO Diagrams
- 3: Extract Test Information from Program Plans: Levels 2 and 3 IPO Diagrams
- 4: Create Test Plan: Levels 2 and 3 IPO Diagrams
- 5: Create Test Design, Test Cases, Test Software, and Test Procedures: Levels 2 and 3 IPO Diagrams
- 6: Perform Formal Test: Levels 2 and 3 IPO Diagrams
- 7: Update Test Documentation: Levels 2 and 3 IPO Diagrams
- 8: Tailoring the Model
- 9: Summing Up

**Appendix A: The Software Engineering Institute, the Capability Maturity Model—Software, and the Capability Maturity Model Integration**

### Appendix B: Preferred Practices

- 1 Program Management Plan and Template
- 2 Software Development Plan and Template
- 3 Software Quality Assurance Plan and Template
- 4 Configuration Management Plan and Template

**Appendix C: Questionnaire for Evaluating Local Testing Processes**

**Appendix D: A Primer for Test Execution**

Glossary

Bibliography

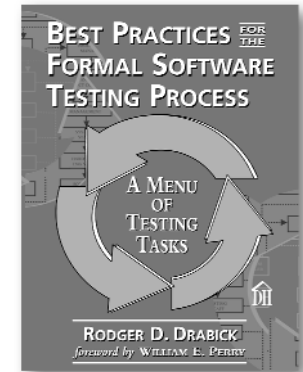
Index

# Best Practices for the Formal Software Testing Process

A Menu of Testing Tasks

by Rodger D. Drabick

foreword by William E. Perry



ISBN: 978-0-932633-58-3  
©2004 312 pages softcover  
\$41.95 (includes \$6 UPS in US)

*A Soup-to-Nuts Series of Tasks for Formal System Testing*

Testing is not a phase. Software developers should not simply throw software over the wall to test engineers when the developers have finished coding. A coordinated program of peer reviews and testing not only supplements a good software development process, it supports it.

A good testing life cycle begins during the requirements elucidation phase of software development, and concludes when the product is ready to install or ship following a successful system test.

Nevertheless, there is no one true way to test software; the best one can hope for is to possess a formal testing process that fits the needs of the testers as well as those of the organization and its customers.

A formal test plan is more than an early step in the software testing process—it’s a vital part of your software development life cycle. This book presents a series of tasks to help

you develop a formal testing process model, as well as the inputs and outputs associated with each task. These tasks include • review of program plans • development of the formal test plan • creation of test documentation (test design, test cases, test software, and test procedures) • acquisition of automated testing tools • test execution • updating the test documentation • tailoring the model for projects of all sizes.

Whether you are an experienced test engineer looking for ways to improve your testing process, a new test engineer hoping to learn how to perform a good testing process, a newly assigned test manager or team leader who needs to learn more about testing, or a process improvement leader, this book will help you maximize your effectiveness.

Read more about this book at [www.dorsethouse.com/books/bpf.html](http://www.dorsethouse.com/books/bpf.html)