

METRICS-DRIVEN ENTERPRISE SOFTWARE DEVELOPMENT

**Effectively Meeting Evolving
Business Needs**

SUBHAJIT DATTA



Copyright ©2007 by Subhajit Datta

ISBN-13: 978-1-932159-64-6

Printed and bound in the U.S.A. Printed on acid-free paper

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data

Datta, Subhajit.

Metrics-driven enterprise software development : effectively meeting evolving business needs / By Subhajit Datta.

p. cm.

Includes index.

ISBN 978-1-932159-64-6 (hardcover : alk. paper)

1. Computer software—Development. 2. Computer software—Quality control.

I. Title.

QA76.76.D47D364 2007

005.3—dc22

2007018289

The following technologies, products, processes, and organizations used in this book may be registered trademarks or service marks: Java, EJB, ASP, JSP, JUnit, JMS, Hewlett-Packard, IBM, Motorola, PSP, TSP, CMM, CMMI, P-CMM, Visual Basic, Struts, AJAX, ISO 9000, ISO 9001, Rational Unified Process, and Unified Software Development Process (aka Unified Process or UP).

This publication contains information obtained from authentic and highly regarded sources. Reprinted material is used with permission, and sources are indicated. Reasonable effort has been made to publish reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

All rights reserved. Neither this publication nor any part thereof may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the publisher.

The copyright owner's consent does not extend to copying for general distribution for promotion, for creating new works, or for resale. Specific permission must be obtained from J. Ross Publishing for such purposes.

Direct all inquiries to J. Ross Publishing, Inc., 5765 N. Andrews Way, Fort Lauderdale, Florida 33309.

Phone: (954) 727-9333

Fax: (561) 892-0700

Web: www.jrosspub.com

DEDICATION

To my parents —
It is amazing how much you taught me without ever teaching.
You also let me be myself.

TABLE OF CONTENTS

List of Tables and Figures	xiii
Preface	xv
About the Author	xix
Web Added Value™	xxi
Chapter 1. Introduction	1
1.1. Overview	1
1.2. What Is This Book About?	1
1.3. Intended Audience	3
1.4. Organization of the Book	5
1.5. Reading Plan	7
1.6. Of I and We, and Other Sensitivities	8
1.7. Why I Wrote This Book	9
1.8. Summary	10
References	10
Chapter 2. Schmoozing with the Ideas	13
2.1. Overview	13
2.2. Ideas and Their Relations	13
2.3. Themes in This Book	15
2.3.1. Enterprise Software Systems	15
2.3.2. Stakeholders	16
2.3.3. Practitioners	17
2.3.4. Measure-Measurement-Metrics	18
2.3.5. Heuristics	19

- 2.3.6. Feedback 21
- 2.3.7. Iterative and Incremental Development 21
- 2.3.8. Process 23
- 2.3.9. Artifacts 24
- 2.4. Threads and Braids 25
- 2.5. Summary 26
- References 27

PART 1. CONTEXT

- Chapter 3. Software Engineering and Metrics 31**
- 3.1. Overview 31
- 3.2. The Soft and Hard of Software Engineering 31
 - 3.2.1. Software Engineering Is Not Supported by Laws of Software, at Least Not Yet 33
 - 3.2.2. Software vis-à-vis a Cheeseburger 33
 - 3.2.3. The Plasticity of Software, or Why a Software Product Is Never Complete 34
 - 3.2.4. The Macro and the Micro States 36
 - 3.2.5. Software Engineering and Communication, or Why Software Engineers Need to Listen 37
 - 3.2.6. We Cannot Build Software the Way We Build Buildings 38
 - 3.2.7. The Open-Source Paradigm of Software Building 38
- 3.3. Software Metrics: A Quick Recce 39
 - 3.3.1. Initial Ideas 40
 - 3.3.2. Deeper Quests 43
 - 3.3.3. Newer Concerns 44
 - 3.3.4. Evolving Odyssey 50
- 3.4. The Need for More 54
- 3.5. Summary 55
- References 55

- Chapter 4. Key Themes in Software Measurements 59**
- 4.1. Overview 59
- 4.2. Many Authors, Many Rubrics 59
- 4.3. The Essence of Software Measurements 60
 - 4.3.1. No Silver Bullet 60
 - 4.3.2. Metrics Hunt in Groups 60
 - 4.3.3. There Will Always Be Assumptions 61
 - 4.3.4. Be Prepared to Customize a Metric 62

4.3.5. Do Not Be Afraid to Create Your Own Metrics	63
4.3.6. Keep It Simple	64
4.3.7. Collect and Compile over Time	65
4.3.8. Use Automation	66
4.3.9. Black Box Is Bad: Be Clear about Scope and Workings	67
4.3.10. Never Use Metrics to Measure People	67
4.3.11. Metrics Give Feedback — The Rest Is Up to You	68
4.4. Summary	69
References	69
Chapter 5. Taking Your Project the Metrics Way	71
5.1. Overview	71
5.2. The Metrics Way	71
5.3. Babbage's Consternation	73
5.4. Metrics and Politics	74
5.5. A Metrics Charter for Your Project	75
5.6. Summary	77
References	78
PART 2. CONSTRUCTS	
Chapter 6. Iterative and Incremental Development: A Brief	
Background	81
6.1. Overview	81
6.2. March of the Methodologies	81
6.3. One Begets the Other	83
6.4. Ceremony and Agility: Twin Concerns	84
6.5. Recurrent Themes: Iteration and Incrementation	85
6.6. The Unified Software Development Process: A Framework	87
6.6.1. Unified Process: Salient Features	88
6.7. The Essence of Our Approach	89
6.8. Summary	90
References	90
Chapter 7. Requirements: The Drivers of Software Development	93
7.1. Overview	93
7.2. The Requirements Realm	93
7.3. Requirements to Use Cases: An Uneasy Transition	94
7.4. Requirements Interaction	96
7.5. Changing Requirements	97

7.6. Growing Our Metrics 98
 7.6.1. Correlation Matrix 98
 7.6.2. Mutation Score 99
7.7. Summary 100
References 100

**Chapter 8. Analysis and Design: On Understanding, Deciding,
and Measuring** 101
8.1. Overview 101
8.2. Analysis and Design or Analysis-Design? 101
8.3. Analysis: The What 102
 8.3.1. Leaving the Baggage Out of Analysis 103
 8.3.2. The Act of Analysis 103
 8.3.3. Analysis Paralysis 108
8.4. Meshing Ratio 111
8.5. Segueing into Action 113
8.6. Abstraction, Elision, and Indirection 114
8.7. AEI Score 119
8.8. Morphing Index 121
 8.8.1. Setting the Stage 121
 8.8.2. Shifting Sands of Design 122
 8.8.3. Making of the Metric 123
 8.8.4. Derivation: First Pass 123
 8.8.5. Derivation: Second Pass 124
 8.8.6. Back to Tina 126
8.9. An Allied Metric: Whitmire’s Volatility Index 127
8.10. Summary 128
References 128

Chapter 9. Implementation: It Takes Some Doing 131
9.1. Overview 131
9.2. Proof of the Pie 131
9.3. The Essence of Implementation 134
9.4. Iterations, Increments, and Implementation 134
9.5. The Challenge of Convergence 135
9.6. Specific Convergence 137
 9.6.1. Deliverable Unit 138
 9.6.2. Risk Factor 139
 9.6.3. Effort Factor 139
 9.6.4. Deriving Specific Convergence 140

9.7. Interface Map: An Artifact for the Foot Soldier	143
9.7.1. Implementation Unit	144
9.7.2. Presenting the Interface Map	146
9.8. Summary	147
References	148
Chapter 10. Testing: Hunt for a Litmus Paper for Software	149
10.1. Overview	149
10.2. The Issue with Testing	149
10.3. Testing Metrics	151
10.4. The Challenge of Testing	152
10.5. Exception Estimation Score	153
10.5.1. Testing Dimensions	154
10.5.2. Deriving the Exception Estimation Score	156
10.5.3. Acting on the Exception Estimation Score	156
10.6. Summary	157
Reference	157
Chapter 11. Putting It All Together	159
11.1. Overview	159
11.2. Getting a “Feel” for the Metrics	159
11.3. The Whole Point — Revisited	160
11.4. A Mechanism for the Metrics	161
11.5. Numbers and Their Meanings	163
11.6. The Road Ahead	164
11.7. Summary	164
References	164
PART 3. CASE STUDY	
Chapter 12. The Plot Thickens	171
12.1. Overview	171
12.2. Background	171
12.3. Tina’s Project	172
12.4. As Things Should Be	174
12.5. And As Things Usually Are	175
12.5.1. Legacy Code	176
12.5.2. Legacy Thinking	177
12.5.3. The Myth of Transitioning	178
12.5.4. Interfacing with Teams from Other Organizations	179
12.5.5. Cross-Continental Teams	179

12.6. Shall the Train Meet? The Story of the Metrics 180
 12.7. Summary 180
 Reference 181

Chapter 13. Getting into the Groove 183
 13.1. Overview 183
 13.2. The First Meeting 183
 13.3. Taking Stock 184
 13.4. Correlation Matrix to the Rescue 186
 13.5. First Pass 188
 13.6. Back and Forth 193
 13.7. Mutation Score: Let the Metric Do the Talking 194
 13.8. Before Development Begins 197
 13.9. Getting on with the Game 200
 13.10. Summary 200

Chapter 14. Diving at the Deep End 203
 14.1. Overview 203
 14.2. What Is So Deep about the Deep End? 203
 14.3. Understanding the Interaction of Requirements 205
 14.4. From Requirements to Analysis: The Meshing Ratio 205
 14.5. From What to How: Getting into Design 208
 14.6. The Guts of Design 212
 14.7. Design Elements 214
 14.8. AEI Score: Of Checks and Balances 216
 14.9. Meshing Ratio: Another Pass 218
 14.10. Keeping the User in the Loop 219
 14.11. Starting to Build in Earnest 220
 14.12. Summary 221
 References 221

Chapter 15. Building Time 223
 15.1. Overview 223
 15.2. Dissecting the Design 223
 15.3. Convergence: Quest for the Holy Grail 231
 15.4. Design Drift 233
 15.5. The Buck Stops Here 246
 15.6. Summary 251
 References 252

Chapter 16. Feel of the World	253
16.1. Overview	253
16.2. What Happened after the Interface Map?	253
16.3. Implementation and Testing	254
16.4. Calculating and Applying the Exception Estimation Score	264
16.5. Summary and Taking Leave	266
References	266
Epilogue	269
Index	271

LIST OF TABLES AND FIGURES

Table 3.1	Software metrics trends
Table 5.1	Sample metrics charter
Table 7.1	Sample <i>Correlation Matrix</i>
Table 8.1	Sample questionnaire for <i>AEI Score</i> calculation
Table 9.1	<i>Risk Factor</i> , <i>Effort Factor</i> , and iteration plans for a project with five <i>Deliverable Units</i>
Table 9.2	<i>Specific Convergence</i> values for Plan A and Plan B
Table 9.3	Sample <i>Interface Map</i>
Table 9.4	<i>Interface Map</i> for <i>Implementation Unit</i> CurrencyController.java
Table 11.1	Metrics and artifacts: a snapshot
Table 13.1	Parts of <i>Correlation Matrix</i> for the existing MAGIC system
Table 13.2	MAGIC 2.0 requirements: first cut
Table 13.3	<i>Mutation Scores</i> for MAGIC 2.0 requirements
Table 13.4	MAGIC 2.0 requirements for first release
Table 13.5	Success criteria for first release of MAGIC 2.0
Table 13.6	MAGIC 2.0 risk list
Table 14.1	Calculation of <i>AEI Score</i>
Table 15.1	<i>Specific Convergence</i> values
Table 15.2	<i>Interface Map</i> for the Contention_Database_Access_Facilitator IU
Table 16.1	Calculation of <i>Exception Estimation Score</i>
Figure 2.1	Interplay of ideas and organization of this book
Figure 6.1	Methodologies in the ceremony-agility space
Figure 9.1	<i>Specific Convergence</i> versus iterations: Plan A and Plan B

- Figure 11.1** Iteration, increment, and the metrics
- Figure 11.2** Varying importance of the metrics across phases on a scale of 0 to 5
- Figure 13.1** Wireframes and navigation paths
- Figure 14.1** *Meshing Ratio*: first pass
- Figure 14.2** *Meshing Ratio*: second pass
- Figure 15.1** *Specific Convergence* versus iterations: Plan A, Plan B, and Plan C
- Figure 15.2** Designing for the contention scenario: first pass
- Figure 15.3** Designing for the contention scenario: second pass

PREFACE

Bell (1937) begins the introduction to his *Men of Mathematics*: “This section is headed *Introduction* rather than *Preface* (which it really is) in the hope of decoying habitual preface-skippers into reading...” I can assure you no such artifice is at play here. This is headed a preface, and this is a preface.

I am an avid reader of prefaces. Fowler (2003) says, “...writing a book is so hard and so compulsive”; prefaces give some hint why someone would want to do it. Authors use different styles for prefaces. Some are diffident — yet another book on so and so subject needs such and such explanations, some defiant — the world needs *this* book, and some distinct — “this book almost seemed to write itself” (Graham et al. 1994). Some say how or why the book came to be; others are filled with anecdotes and wisecracks.

All who write a book for fun should agree that bookmaking is an organic experience. (I am not sure, though, why *bookmakers* are dubiously linked with games of chance but *filmmakers* and *homemakers* have exalted pursuits.) The act of writing is just a culmination; thoughts and ideas germinate long before they come to the finger. But only when one writes do many of these crystallize. I thought about a book like this for a long time, but not until I started writing did I fully feel my thoughts.

Software engineering is a young discipline. It has been thousands of years that humans have built magnificent buildings, and clever devices of locomotion have been around for a few hundred years, but only a few decades that software has been built on a scale that affects our lives in powerful and profound ways. Booch (2006) calls it “...a tremendous privilege as well as a deep responsibility to be a software developer.” Much of the wisdom of building smart, efficient, and beautiful software still lies buried in people’s heads and in folklore of the

trade. I joined the software industry in the twilight of the last millennium. I have since marveled at the scope of writing software engineering presents. We are still searching for first principles and basic laws, even as we build more and more complex systems: the accompanying trepidation, tension, and triumph are the stuff of literature.

This is a practice-oriented book, meant to share experience, observation, and insight with other practitioners. This book presents a set of metrics to help make enterprise software systems easier to develop and smoother to enhance. The ideas and their applications are simple; you might think, “I could have thought of that.” I hope you do, as that is the whole point. Only when we feel we can do something do we actually want to try it out. I want you to try out the ideas in this book, make them your own, modify and improve upon them. I have applied many of the ideas and have lived to tell the tale! Chapter 1 goes deeper into the theme of the book, provides a reading plan, and discusses who should read it (if you have come this far, let me assure you that *you* should), among other topics. Writing this book sharpened my own sense of what I know about the subject and made me rewind and relearn some of the much I do not.

Writing the book has been exciting. It took me back to another book-writing project of which I was partly a part, what seems eons ago. My father was an engineer, an accountant, and a man of many other interests. We always thought he could write books. A few years after he had successfully set up his own management consultancy practice, he finally got down to writing one. It was titled *ISO 9000: A Roadmap for Design, Installation and Implementation of Quality Management Systems* and was accepted by a leading publisher. As he wrote the book, he often asked my opinion on one aspect or another. I was all too eager to offer opinions. A sophomore, I felt equal to commenting on his style of writing and organization of the chapters. After submitting the manuscript, my father was admitted to a hospital for treatment of some minor symptoms. He asked me to borrow a few books for him from the local British Council Library to help plan his next book as he “served out” the hospital time. I did. Three weeks later, I sifted through the pages of those books, pulled out the bookmarks, and returned them to the library. My father was no more.

Any reflective work takes us back to our roots. I share my enthusiasm for this book with my mother — who is both a mother and a father now. My brothers have always remained for me acmes of original thinking and sources of much encouragement.

Most of this book was written during the first year of my marriage. A distracted husband is no recipe for early nuptial bliss; that bliss abounds goes to the credit of Reshmi — a very old friend and a very new wife. She gave me love, understanding, and sumptuous food and took away the cares of functioning.

I wrote this book while we lived in Tallahassee. Tallahassee, in addition to Florida sunshine, offers a delicious detachment. But my unsocial grain has not gained me fully, only because Dr. Tapas Bhattacharyya is here. He gave me enriching company, use of his laser printer, and took the photo for the author biography. In general, being mired in this book has helped me to take life's major and minor irks with greater humor.

I make it a point to forget the origins of best ideas I come across. Then it is easy to believe they were originally mine. This habit prevents a list of acknowledgments. I hope originators of best ideas are too deep into worthwhile things to mind.

Whatever the reader finds sublime or useful in this book cannot be claimed to be mine; it comes from the chemistry of circumstances shaping my life. Whatever the reader finds feckless speaks of my inability to make the best of what came my way.

Along the lines of what the pilot says as the plane taxis toward the arrival gate, I understand you had many choices but picked up this book, and I appreciate your indulgence (actually the pilot says *business*, but I am chary of sounding too gross). Thank you for trying out *Metrics-Driven Enterprise Software Development: Effectively Meeting Evolving Business Needs*.

Subhajit Datta
Tallahassee, Florida
subhajit.datta@gmail.com

REFERENCES

- Bell, E. T. (1937). *Men of Mathematics*. Simon and Schuster.
- Booch, G. (2006). *Handbook of Software Architecture*. <http://www.booch.com/architecture/index.jsp>.
- Fowler, M. (2003). *Patterns of Enterprise Application Architecture*. Addison-Wesley.
- Graham, R. L., Knuth, D. E., and Patashnik, O. (1994). *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley.

ABOUT THE AUTHOR



Subhajit Datta has a master of science in computer science and an undergraduate degree in electrical engineering. He is an IBM Certified Specialist for Rational Unified Process v2003 and has additional certification in object-oriented analysis and design with Unified Modeling Language (IBM Test 486).

Mr. Datta has wide experience in developing enterprise software systems. He worked as a software engineer with IBM Global Services in roles of technical lead, business systems analyst, release manager, and application developer. One of his papers was published as intellectual capital in the IT Process Model knowledge network of IBM's ICM AssetWeb.

Mr. Datta is currently working on his doctoral dissertation at Florida State University. His research interests include software architecture, design tools and techniques, metrics, and requirements, and he has several research publications. He is also involved in teaching courses on Internet technologies and Java.

Mr. Datta is a member of Upsilon Pi Epsilon, the international honor society in the computing and information disciplines, the Institute of Electrical and Electronics Engineers (IEEE), and the Association for Computing Machinery (ACM).

Reading, writing, and travel are among Mr. Datta's interests. He may be reached at subhajit.datta@gmail.com.



*Free value-added materials available from
the Download Resource Center at www.jrosspub.com*

At J. Ross Publishing we are committed to providing today's professional with practical, hands-on tools that enhance the learning experience and give readers an opportunity to apply what they have learned. That is why we offer free ancillary materials available for download on this book and all participating Web Added Value™ publications. These online resources may include interactive versions of material that appears in the book or supplemental templates, worksheets, models, plans, case studies, proposals, spreadsheets and assessment tools, among other things. Whenever you see the WAV™ symbol in any of our publications, it means bonus materials accompany the book and are available from the Web Added Value Download Resource Center at www.jrosspub.com.

Downloads available for *Metrics-Driven Enterprise Software Development: Effectively Meeting Evolving Business Needs* consist of a metrics-driven software development presentation and a quick reference glossary of important metrics.

