MASTERING SOFTWARE PROJECT MANAGEMENT

Best Practices, Tools and Techniques

SKN008759

MURALI CHEMUTURI
THOMAS M. CAGLEY, JR.

005.1
C517

MASTERING SOFTWARE PROJECT MANAGEMENT

Best Practices, Tools and Techniques

MURALI CHEMUTURI
THOMAS M. CAGLEY, JR.

THƯ VIỆN TRƯỜNG ĐHSPKT
SKN 008759

J.ROSS PUBLISHING

Copyright ©2010 Copyright J. Ross Publishing

ISBN-13: 978-1-60427-034-1
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
Chemuturi, Murali, 1950-
Mastering software project management : best practices, tools and techniques / by Murali Chemuturi and Thomas M. Cagley Jr.
p. cm.
Includes index.
ISBN 978-1-60427-034-1 (hardcover : alk. paper)
1. Computer software--Development. 2. Project management. I. Cagley, Thomas M., 1956- II. Title.
QA76.76.D47C478 2010
005.1--dc22 2010019185

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, FL 33309.

Phone: (954) 727-9333
Fax: (561) 892-0700
Web: www.jrosspub.com

CONTENTS

Foreword ix
Preface xiii
About the Authors xv
Web Added Value™ xvii

Chapter 1. Software Project Basics 1
Introduction 1
Types of Software Projects 5
Classifications of Software Projects 6
Based on Software Development Life Cycle 7
Approach Driven 9
Maintenance 12
Web Application 16
Agile Development 17
Conclusion 17

Chapter 2. Approaches to Software Project Management 19
Alignment of Software Engineering Methodology with Project Management Methodology 19
The Ad Hoc Methods-Based Approach 21
The Process-Driven Approach 22
So, What *Is* the Right Approach? 23
The Ad Hoc Approach 24
The Process-Driven Approach 24
But Is a Process-Driven Approach the Right Choice? 24
In a Process-Driven Approach: What Process and How Much? 26

Chapter 3. Software Project Acquisition 31
From an External Client 31

The Request for Proposal 32
The Proposal 34
Negotiation 42
Contract Acceptance 43
From an Internal Client 44
The Feasibility Study 45
Preparing the Proposal 47
Finalizing the Proposal 47
Reference 48

Chapter 4. Software Project Initiation 49
Introduction 49
Initiation Activities 49
Project Management Office-Level Activities 52
Identifying the Software Project Manager 52
Preparing/Handing Over the Project Dossier to the Software Project Manager 52
Coordinating Allocation of Project Resources 53
Assisting the Software Project Manager in Obtaining Necessary Service Level Agreements from Departments in the Organization 55
Assisting the Software Project Manager with the Project Kickoff Meeting 55
Software Project Manager-Level Activities 55
Ensuring that Project Specifications Are Complete 57
Reviewing Estimates and Revisions/Updates of Estimates 57
Identifying Necessary Resources and Raising Requests 59
Preparing Project Plans 62
Setting Up the Development Environment 63
Arranging for Project-Specific Skill Training 63
Organizing the Project Team 64
Training the Project Team on the Project Plans 64
Conducting a Project Kickoff Meeting 65
Arranging for a Phase-End Audit 65
Common Pitfalls in Project Initiation 66
Identifying the Wrong Software Project Manager 66
Identifying Inappropriate Resources 66
Incurring Delays in Software Project Initiation Activities 67
References 67

Chapter 5. Software Project Planning **69**
Introduction 69
Planning Defined 71
Plans Prepared in Software Project Management 73
The Project Management Plan 77
Resources 77
Skill Sets 77
Computer Systems 79
Project Management Method 79
The Configuration Management Plan 80
Naming Conventions 83
Change Management 84
The Quality Assurance Plan 85
The Schedule Plan 86
The Induction Training Plan 86
The Risk Management Plan 87
The Build Plan 88
The Deployment Plan 88
The User Training Plan 89
The Handover Plan 90
The Software Maintenance Plan 90
The Documentation Plan 90
Roles in Planning 91
The Organization 91
The Software Project Manager 92
Pitfalls in Software Project Planning 93
Best Practices in Software Project Planning 95
References 96

Chapter 6. Software Project Execution **97**
Introduction 97
Work Management 98
Work Registers 100
De-allocation 102
Configuration Management 104
Information Artifacts 104
Code Artifacts 106
Configuration Registers 111
Configuration Management Tools 115
Quality Management 117

Verification Techniques 119
Validation Techniques 120
Product Testing 121
Allocation of Quality Assurance Activities 124
But *How Much* Quality Assurance? 124
Testing Tools 125
Morale Management 126
Motivation 126
Conflict 130
Productivity Management 131
Stakeholder Expectations Management 133
Product Integration Management 138
Pitfalls and Best Practices 140

Chapter 7. Software Project Execution Control 143
Introduction 143
Aspects of Control in Project Execution 144
Scope Control 145
Cost Control 146
Schedule/Progress Control 147
Quality Control 147
Effort Control 148
Productivity Monitoring 148
Control Mechanisms 149
Progress Assessment: Earned Value Analysis 153

Chapter 8. Change Management in Software Development Projects 157
Introduction 157
Origins of Change 158
The Change Request Register 160
Change Request Resolution 162
Change Request Implementation Strategy 163
The Value of Metrics Derived from a Change Request Register 167

Chapter 9. Scheduling 171
Introduction 171
The Initial Work Breakdown Structure 172
A Work Breakdown Structure with Predecessors Defined 172
A Work Breakdown Structure with Initial Dates 176
A Work Breakdown Structure with Resource Allocation 178
Scheduling in Practice 181
Graphic Representation of a Schedule 181

Chapter 10. Software Project Closure **183**
Introduction 183
Identifying Reusable Code Components 185
Documenting the Best Practices 186
Documenting the Lessons Learned 187
Collecting/Deriving and Depositing the Final Project Metrics in the Organizational Knowledge Repository 188
Conducting Knowledge-Sharing Meetings with Peer Software Project Managers 188
Depositing Project Records with the Project Management Office 189
Depositing Code Artifacts in the Code Repository 190
Conducting the Project Postmortem 190
Releasing the Software Project Manager 191
Closing the Project 192
The Role of the Organization in Project Closure 192
The Project Management Office 192
The Configuration Control Board 193
The Systems Administration Department 194
Reference 194

Chapter 11. Agile Project Management **195**
Introduction 195
Project Management Roles 195
Agile Project Management Characteristics 196
Metaphor 197
Teamwork and Collaboration 197
Guiding Principles 198
Open Information 198
Use a Light Touch 199
Monitoring and Adjustment 199
The Nuts and Bolts of Agile Project Management 200
Planning the Work 200
Controlling the Work 202
Process Improvement 205
Reference 206

Chapter 12. Pitfalls and Best Practices in Software Project Management **207**
Introduction 207
Organizational-Level Pitfalls and Best Practices 208
Process-Driven Project Management 208

An Ineffective Project Management Office or No Project Management Office 208
Poor Project Initiation 210
Poor Software Estimation 211
Poor Project Planning 211
The Wrong Service Level Agreements 212
Poor Standards and Guidelines for Software Development 213
Poor Project Oversight 214
Inadequate Project Management Training 214
Software Project Manager-Level Pitfalls and Best Practices 215
Fair Treatment of Project Human Resources 216
A Balanced Workload 216
Equitable Rewards 217
Poor Software Estimation 217
Poor Project Planning 217
Informal Issue Resolution 217
Poor Change Management 218
Poor Record Keeping 218
Additional Best Practices for Software Project Management 219
A Knowledge Repository 219
Continuous Process Improvement 219
Project Postmortems 219
Training in the Soft Skills 220
Information Sharing 220
Management Support 220
Some Closing Words 221

Appendix A. Management of Software Development Projects 223

Appendix B. Decision-Making for Software Project Managers 237

Appendix C. People Management 251

Appendix D. Productivity Concepts for Software Project Managers 271

Appendix E. Issue Resolution in Software Project Management 287

Appendix F. Measurement and Metrics in Software Development Organizations 295

Appendix G. Measurement and Management of Customer Satisfaction 315

Appendix H. An Introduction to PERT/CPM 327

Appendix I. Abbreviations 347

Appendix J. Templates for Software Project Managers 351

Index 373

FOREWORD

When a software project fails, the impact to an organization can be significant. The losses the organization experiences may be financial, the project's failure may negatively impact the organization's credibility with customers, or the organization may lose market position and competitive advantage. As of this writing, the latest statistics from the 2009 Standish Group CHAOS report state that "44% were challenged which are late, over budget, and/or with less than the required features and functions." This is an alarming number. It represents a dismal trend and is in fact an increase in the number of failures, representing the highest failure rate in over a decade.

Although the reasons for these failures vary, there is every reason to believe that some of these failures are directly attributable to ineffective software project management. Furthermore, it is safe to assume that in nearly all of these failures, project management played at least a minor role. Show me a failed project and I will show you a failed project manager or process.

Does the blame for a failed project always rest on the project manager's shoulders? Can organizational behaviors be blamed for some of these failures? Can a customer or client cause a project to fail? If we concede that, from time to time, there are some extraordinary causes of failure that are outside the boundary of a project manager's control, plenty of fingers are still left pointing at ineffective project management skills, practices, techniques, and technologies.

So, what is at the heart of these project failures? Is it the skill level of the individual project manager? Is it the result of unavailable or inaccurate project data? Is it a lack of training or a lack of a proper project management practices?

To state the obvious, it is essential that software development organizations focus their attention on improving the state of their project management practices. More to the point, there is enough collective experience and understanding that we can now acknowledge and identify software project management best practices.

This is in fact what Murali Chemuturi and Tom Cagley have accomplished in *Mastering Software Project Management: Best Practices, Tools and Techniques.* They have compiled their collective experience and wisdom and have presented it in a practical, clear, and concise format that is of value to a new project manager as well as more experienced project managers. The authors present the process and the philosophy behind the methodologies and techniques of software project management.

The introduction to the book quickly puts context around the authors' working definition of a software development project and the classification of a software project, which includes the software development life cycle, approach-driven software, software maintenance, agile software development, and Web-application development. The authors are clear — this book addresses software project management utilizing a process-driven approach.

Their project management framework comprises four core elements: acquisition, initiation, execution, and closure. In the four chapters that discuss these core elements, we learn about various workflows and best practice activities and are given helpful tools and techniques. The book is formatted in such a way as to be useful as an educational text as well as a practical reference guide for the practitioner.

Mastering software project management is about gaining control through the use of reliable and meaningful information. The authors have addressed the importance of control with insightful chapters on planning, scheduling, and change management. Once again, they present process flows, useful templates, and guidance on how to execute the best practices they have outlined.

Software project management best practices would not be complete without a chapter on agile development. The authors drive home the point that for an organization to achieve success, the organization must first embrace the agile philosophy. Here, too, the authors discuss how to plan and control the workflow in an agile environment.

Chapter 12 discusses pitfalls and best practices and begins with an important observation regarding how best practices may be viewed in an organization. The authors point out that management is unique to an organization based on that organization's culture. And while producing positive results is an important aspect of management, of greater importance is that the results are achieved with optimal costs, productivity, quality, and morale. In their words, "... it is necessary for organizations to adopt best practices and avoid the pitfalls to grow and prosper."

And perhaps, quite possibly, the authors have saved the best for last. There are ten appendixes that focus on project management intangibles, such as Appendix C, dealing with the art and science of managing people, in which the authors discuss such topics as delegating and managing your superiors. Appendix

B addresses the practice of making decisions. Appendix E centers on issue resolution. The list goes on to include other relevant topics such as productivity, measures and metrics, and customer satisfaction. Yes, all of these topics should be considered to be part of a comprehensive discussion on mastering software project management.

This book should be required reading for all software project managers and for those organizations that struggle with late and costly project implementation. The advice and processes so rigorously detailed in this book should become the de facto standard for improving an organization's development practices.

— **David Herron**
Vice President of Knowledge Solution Services
David Consulting Group, USA

PREFACE

Before we talk about software project management, we need to understand the terms *project* and *management*. Once we understand these two terms and how they relate to each other, we can apply them to the field of software development. In the subsequent chapters and appendices, we strive to define and explain and then to show just what software project management is.

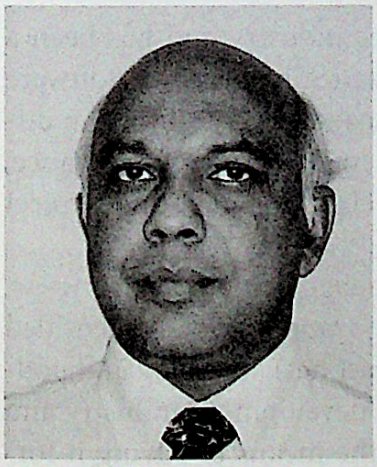
Software projects are initiated, planned, and executed at two places. The first is within the same organization that is going to use the end product; the second is in an organization specialized in developing software that will be used by other organizations. In this book, we are not considering hobby projects although they can use the techniques in the book also. Projects can also be called internal projects and external projects. Internal projects are those with an end product that is used within the same organization; external projects are those with an end product that is used by another organization. We will expand on these terms in subsequent chapters.

Our focus in this book is on the management of software projects that are tightly planned, monitored, controlled, and executed. These projects could be internal or external projects.

This book is not geared to help you pass exams of any kind or to receive certification. Rather it helps you pass the examination conducted by life and to achieve success in software project management. This book is based on our experiences: experiences gained through academic efforts, reading books and literature, and managing and observing real-life projects. What we have learned has been plowed back in to fine tune our theories. This book is a result of that fine tuning and is something akin to the expression "What they don't teach you at Harvard."

We will be glad to receive your feedback and promise to usually respond in one business day. We can be reached at murali@chemuturi.com and t.cagley@davidconsultinggroup.com.

ABOUT THE AUTHORS

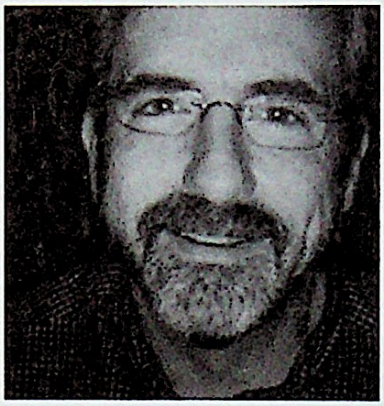


Murali Chemuturi is an information technology and software development subject matter expert, hands-on programmer, author, consultant, and trainer. In 2001, he formed Chemuturi Consultants, his own IT consulting and software development firm. Chemuturi Consultants help software development organizations to achieve their quality and value objectives. The firm provides training in several software engineering and project management topics, such as software estimation, test effort estimation, function point analysis, and software project management, to name a few. Chemuturi Consultants also offers a number of products to aid project managers and software development professionals, such as PMPal, a software project management tool, and EstimatorPal, FPAPal, and UCPPal, a set of software estimation tools.

Prior to starting Chemuturi Consultants, Murali gained over 15 years of industrial experience in various engineering and manufacturing management positions. He then gained over 24 years of information technology and software development experience. His most recent position prior to forming Chemuturi Consultants was vice president of Software Development at Vistaar e-Business Pvt., Ltd.

His undergraduate degrees and diplomas are in electrical and industrial engineering. He holds a MBA and a postgraduate diploma in computer methods and programming. Murali also has several years of academic experience, teaching a variety of computer and IT courses, such as COBOL, Fortran, BASIC, computer architecture, and database management systems.

He is a member of IEEE, a senior member of the Computer Society of India, and a Fellow of the Indian Institute of Industrial Engineering as well as a well-published author in professional journals.



Thomas M. Cagley, Jr. is vice president of consulting for the David Consulting Group. He is an authority in guiding organizations through the process of integrating software measurement with model-based assessments to yield effective and efficient process improvement programs. He is a recognized industry expert in the measurement and estimation of software projects. His areas of expertise encompass management experience in project management, development methods and metrics, quality integration, quality assurance, and the application of the Software Engineering Institute's Capability Maturity Model Integration® to achieve process improvements.

Tom has over 20 years experience in the software industry and has been a consultant since 1997. He was previously metrics practice manager at Software Productivity Research. Earlier, he held technical and managerial positions in different industries as a leader in software methods and metrics, quality assurance, and systems analysis. He is a frequent speaker at metrics, quality, and project management conferences.

Tom blogs about software development and management topics at www.tcagley.wordpress.com and edits a podcasts of essays and interviews entitled Software Process and Measurement Cast (www.spamcast.net). The blog and podcast serve as a platform to share information with the development industry and to foster a continuing dialog. Sharing and dialog are the means of paying it forward.

He is a member of the International Function Point Users Group and is a Certified Function Point Specialist. He is currently the immediate past president of IFPUG and previously served as the chair of the IFPUG Conference Committee and director of Conferences and Education.

Tom has a B.S. from Louisiana State University and has done extensive postgraduate work at Cleveland State University, Case Western Reserve University, and Kent State University.

Web
Added
Value™

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 *Mastering Software Project Management: Best Practices, Tools and Techniques* consist of free software project planning templates; a demo version of an integrated and collaborative software project managment tool called PMPal, which is valid for 180 days; TestPal, a free, fully functional tool useful for test effort estimations and test case design; and PET, a free personal effectiveness tool.

THƯ VIỆN TRƯỜNG ĐHSPKT
SKN 008759

"**Mastering S[illegible] Project Management** is a masterpiece of clarity, organization and [illegible] practical knowledge. Read it thoughtfully and with care as it is a rich [illegible] trove of experience and insight. Remember the old saying 'don't w[illegible] work smart'—this book will give you the knowledge and tools to d[illegible] that."

—**Robert C. Anderson**, Director, Process Development and Quality Assurance, Computer Aid, Inc.

"This book provides you with tools, techniques, and practices that you can use immediately on your own projects. It supplies the information you need to diagnose your team's situation and presents practical advice to help you build better software—a must read!"

—**John Zielinski**, Project Manager, IBM Global Services

Written from the standpoint of a project manager working in a software development organization, this unique guide explains software project management in its entirety, including project acquisition and execution with backward linkages to concepts that play a facilitation role in successful project management. It provides all the guidance, best practices, tools and techniques needed to master software project management and achieve superior results.

KEY FEATURES

- Presents best practices and pitfalls to avoid for nearly every topic discussed
- Details management of work, configuration, quality management, productivity, team morale and expectations management of the project team, functional management, senior leadership and customers
- Covers all facets of planning—project management, induction training, and deployment—and every aspect of project control
- Explains decision-making concepts from the perspective of software project managers and includes all the tools and techniques needed for making quantitative decisions
- WAV offers downloadable software project planning templates, a demo version of a collaborative software project management tool (PMPal), a tool useful for test effort estimations and test case design (TestPal); and a personal effectiveness tool (PET)—**available from the Web Added Value™ Download Resource Center at www.jrosspub.com**

JR0341

J.ROSS PUBLISHING
www.jrosspub.com

WAV Web Added Value™
Downloadable Resources at www.jrosspub.com

ISBN: 978-1-60427-034-1
90000
9 781604 270341