The Economics of Software Quality

Capers Jones
Olivier Bonsignour

Foreword by Thaddeus Arroyo
Chief Information Officer, AT&T Services, Inc.
Praise for *The Economics of Software Quality*

“This book provides the best treatment on the subject of economics of software quality that I’ve seen. Peppered with valuable industry data, in-depth analysis, empirical methods for quality improvement, and economic analysis of quality, this book is a must-read for anyone who is interested in this subject. With the many real-life and up-to-date examples and stories linking software quality to daily-life activities, readers will find this book an enjoyable read.”

—Stephen H. Kan, Senior Technical Staff Member and Program Manager, Software Quality—IBM Systems and Technology Group, and author of *Metrics* and Models in Software Quality Engineering

“Finally, a book that defines the cost and economics of software quality and their relationship to business value. Facts such as the inability of testing alone to produce quality software, the value of engineering-in quality, and the positive ROI are illustrated in compelling ways. Additionally, this book is a must-read for understanding, managing, and eliminating ‘technical debt’ from software systems.”

—Dan Galorath, CEO, Galorath Incorporated & SEER by Galorath

“Congrats to Capers and Olivier as they release their relevant, extensive, and timely research on the costs of defects in today’s software industry. The authors don’t stop with the causes of defects; they explore injection points, removal, and prevention approaches to avoid the ‘technical mortgage’ associated with defective software products. In today’s ‘quick-to-market’ world, an emphasis on strengthening the engineering in software engineering is refreshing. If you’re a software developer, manager, student, or user, this book will challenge your perspective on software quality. Many thanks!”

—Joe Schofield, Sandia National Laboratories; Vice President, IFPUG; CQA, CFPS, CSMS, LSS BB, SEI-certified instructor

“Whether consulting, working on projects, or teaching, whenever I need credible, detailed, relevant metrics and insights into the current capabilities and performance of the software engineering profession, I always turn to Capers Jones’s work first. In this important new book, he and Olivier Bonsignour make
Table of Contents

Contents
Foreword
Preface
Acknowledgments
About the Authors
Chapter 1: Defining Software Quality and Economic Value
   Introduction
     Why Is Software Quality Important?
   Defining Software Quality
   Defining Economic Value and Defining the Value of Software Quality
     The Economic Value of Software and Quality to Enterprises that Build Internal Software for Their Own Use
     The Economic Value of Software and Quality to Internal Software Users
     The Economic Value of Software and Quality to Commercial Software Vendors
     The Economic Value of Software and Quality to COTS Users and Customers
     The Economic Value of Software and Quality to Embedded Software Companies
     The Economic Value of Software and Quality to Embedded Equipment Users
# Table of Contents

The Economic Value of Software and Software Quality to Other Business Sectors  
Multiple Roles Occurring Simultaneously  
Summary and Conclusions  
Chapter 2: Estimating and Measuring Software Quality  
Introduction  
Using Function Point Metrics for Defect Potentials  
Software Defect Potentials  
  The Special Case of Software Requirements  
  The Origins of Software Requirements  
  The Size, Structure, and Completeness of Software Requirements  
  Minimizing Software Requirements Defects  
  Conclusions about Software Requirements Defects  
  The Special Case of Coding Defects  
Estimating Software Defect Prevention  
Estimating Software Defect Detection and Defect Removal  
Measuring Application Structural Quality  
  Measuring Reliability  
  Measuring Performance Efficiency  
  Measuring Security  
  Measuring Maintainability  
  Measuring Size  
  Summary of Application Structural Quality Measurement Attributes  
Examples of Structural Quality Assessments  
  Bypassing the Architecture  
  Failure to Control Processing Volumes  
  Application Resource Imbalances  
  Security Weaknesses  
  Lack of Defensive Mechanisms
Table of Contents

Desiderata for Systems Evaluating Structural Quality

Three Problems That Distort Software Economic Analysis
  Leakage from Software Historical Data
  Economic Problems with Lines of Code (LOC) Metrics
  Economic Problems with Cost-per-Defect Metrics
  Case A: Poor Quality
  Case B: Good Quality
  Case C: Zero Defects
  Useful Rules of Thumb for Predicting Software Defect Potentials

Summary and Conclusions on Software Quality Estimation and Measurement

Chapter 3: Software Defect Prevention
  Introduction
    The Early History of Defect Prevention Studies in the 1970s at IBM
    Synergistic Combinations of Defect Prevention Methods
    Defect Potentials and Defect Origins
    Defect Prevention, Patterns, and Certified Reusable Materials
    Software Defect Prevention and Application Size

Analysis of Defect Prevention Results
  Agile Embedded Users
  Automated Quality Predictions
  Benchmarks of Software Quality Data
  Capability Maturity Model Integrated (CMMI)
  Certification Programs
  Cost-per-Defect Measures
  Cost of Quality (COQ)
  Cyclomatic Complexity Measures (and Related Complexity Measures)
  Defect Measurements and Defect Tracking
  Formal Inspections
  Function Point Quality Measures
Table of Contents

ISO Quality Standards, IEEE Quality Standards, and Other Industry Standards
Quality Function Deployment (QFD)
Risk Analysis
Six Sigma
Static Analysis

Summary and Conclusions of Software Defect Prevention

Chapter 4: Pretest Defect Removal

Introduction
Small Project Pretest Defect Removal
Large System Pretest Defect Removal
Analysis of Pretest Defect Removal Activities
  Personal Desk Checking
  Informal Peer Reviews
  Automated Text Checking for Documents
  Proofs of Correctness
  Scrum Sessions
  Poka Yoke
  Kaizen
  Pair Programming
  Client Reviews of Specifications
  Independent Verification and Validation (IV&V)
Software Quality Assurance (SQA) Reviews
Phase Reviews
  Inspections (Requirements, Architecture, Design, Code, and Other Deliverables)
  User Documentation Editing and Proofreading
  Automated Static Analysis of Source Code

Summary and Conclusions about Pretest Defect Removal

Chapter 5: Software Testing
Table of Contents

Introduction
   Black Box and White Box Testing
   Functional and Nonfunctional Testing
   Automated and Manual Testing
   Discussion of the General Forms of Software Testing
   Subroutine Testing
   PSP/TSP Unit Testing
   Extreme Programming (XP) Unit Testing
   Unit Testing
   New Function Testing
   Regression Testing
   Integration Testing
   System Testing
   The Specialized Forms of Software Testing
   Stress or Capacity Testing
   Performance Testing
   Viral Protection Testing
   Penetration Testing
   Security Testing
   Platform Testing
   Supply Chain Testing
   Clean Room Testing
   Litigation Testing
   Cloud Testing
   Service Oriented Architecture (SOA) Testing
   Independent Testing
   Nationalization Testing
   Case Study Testing
   The Forms of Testing Involving Users or Clients
   Agile Testing
   Usability Testing
Table of Contents

Field Beta Testing
Lab Testing
Customer Acceptance Testing
Test Planning
Test Case Design Methods
Errors or Bugs in Test Cases
Numbers of Testing Stages for Software Projects
Testing Pattern Variations by Industry and Type of Software
Testing Pattern Variations by Size of Application
Testing Stages Noted in Lawsuits Alleging Poor Quality
Using Function Points to Estimate Test Case Volumes
Using Function Points to Estimate the Numbers of Test Personnel
Using Function Points to Estimate Testing Effort and Costs
Testing by Developers or by Professional Test Personnel

Summary and Conclusions on Software Testing

Chapter 6: Post-Release Defect Removal

Introduction
  Post-Release Defect Severity Levels
  Severity Levels from a Structural Quality Perspective
  Maintainability of Software
  Defect Discovery Rates by Software Application Users
  Invalid Defect Reports
  Abeyant Defects That Occur Under Unique Conditions
  Duplicate Defects Reported by Many Customers
  First-Year Defect Discovery Rates
  Measuring Defect Detection Efficiency (DDE) and Defect Removal Efficiency (DRE)
  Variations in Post-Release Defect Reports
  Variations in Methods of Reporting Software Defects
  Who Repairs Defects after They Are Reported?
# Table of Contents

Case Study 1: Development Personnel Tasked with Maintenance Defect Repairs  
Case Study 2: Maintenance Specialists Handle Defect Repairs  
Comparing the Case Studies  
Litigation Due to Poor Quality  
Cost Patterns of Post-Release Defect Repairs  
Software Occupation Groups Involved with Defect Repairs  
Examining the Independent Variables of Post-Release Defect Repairs  
The Size of the Application in Function Points  
Error-Prone Modules in Software Applications  
User and Industry Costs from Post-Release Defects  
Impact of Security Flaws on Corporations and Government Agencies  
Customer Logistics for Defect Reports and Repair Installation  
Case Study 1: A Small Application by a Small Company  
Case Study 2: A Large Application by a Large Company  
Measurement Issues in Maintenance and Post-Release Defect Repairs  

Summary and Conclusions on Post-Release Defects  

Chapter 7: Analyzing the Economics of Software Quality  

Introduction  
The Economic Value of Software  
Methods of Measuring Value  
Funding Approval and Application Size  
The Impact of Software Construction Difficulties on Software Quality  
Revenue Generation from Software  
Difference Between Software and Other Industries  
Cost Reduction from Software  

Economic Impact of Low-Quality and High-Quality Software  
Software Development and Maintenance  
Software as a Marketed Commodity
# Table of Contents

Software as a Method of Human Effort Reduction  
Software and Innovative New Kinds of Products  
Technical DebtA Measure of the Effect of Software Quality on  
Software Costs  
A Framework for Quantifying Business Value  
Moving Beyond Functional Quality  
The Impact of Software Structure on Quality  
The Impact of Staff Training on Quality  
The Impact of Professional Certification on Quality  
The Impact of Technology Investment on Quality  
The Impact of Project Management on Quality  
The Impact of Quality-Control Methodologies and Tools on Quality  
The Impact of High and Low Quality on Software Schedules  
The Impact of High and Low Quality on Software Staffing  
The Impact of High and Low Quality on Software Development Effort  
The Impact of High and Low Quality on Development Productivity Rates  
The Impact of High and Low Quality on Software Development Costs  
The Impact of High and Low Quality on Development Cost per Function Point  
The Impact of High and Low Quality on Project Cancellation Rates  
The Impact of High and Low Quality on the Timing of Cancelled Projects  
The Impact of High and Low Quality on Cancelled Project Effort  
The Impact of High and Low Quality on Effort Compared to Average Projects  
The Impact of High and Low Quality on Software Test Stages  
The Impact of High and Low Quality on Testing as a Percent of Development  
The Impact of High and Low Quality on Test Cases per Function Point
Table of Contents

The Impact of High and Low Quality on Numbers of Test Cases Created
The Impact of High and Low Quality on Test Coverage
The Impact of Professional Testers on High and Low Quality
The Impact of High and Low Quality on Software Defect Potentials
The Impact of High and Low Quality on Total Software Defects
The Impact of High and Low Quality on Defect Detection Efficiency (DDE)
The Impact of High Quality and Low Quality on Defect Removal Efficiency (DRE)
The Impact of High and Low Quality on Total Defect Removal
The Impact of High and Low Quality on Defects Delivered to Customers
The Impact of High and Low Quality on Delivered Defects per Function Point
Impact of High and Low Quality on Delivered Defect Severity Levels
The Impact of High and Low Quality on Severe Defects per Function Point
The Impact of High and Low Quality on Software Reliability
The Impact of High and Low Quality on Maintenance and Support
The Impact of High and Low Quality on Maintenance and Support Costs
The Impact of High and Low Quality on Maintenance Defect Volumes
The Impact of High and Low Quality on Software Enhancements
The Impact of High and Low Quality on Enhancement Costs
The Impact of High and Low Software Quality on Maintenance and Enhancement Staffing
The Impact of High and Low Quality on Total Effort for Five Years
The Impact of High and Low Quality on Total Cost of Ownership (TCO)
The Impact of High and Low Quality on Cost of Quality (COQ)
# Table of Contents

The Impact of High and Low Quality on TCO and COQ per Function Point
The Impact of High and Low Quality on the Useful Life of Applications
The Impact of High and Low Quality on Software Application Tangible Value
The Impact of High and Low Quality on Return on Investment (ROI)
The Impact of High and Low Quality on the Costs of Cancelled Projects
The Impact of High and Low Quality on Cancellation Cost Differentials
The Distribution of High-, Average-, and Low-Quality Software Projects

Summary and Conclusions on the Economics of Software Quality
High-Quality Results for 10,000 Function Points
Low-Quality Results for 10,000 Function Points

References and Readings
Index