

Practical Software Metrics For Project Management And Process Improvement

Based on the needs of the educational community, and the software professional, this book takes a unique approach to teaching software testing. It introduces testing concepts that are managerial, technical, and process oriented, using the Testing Maturity Model (TMM) as a guiding framework. The TMM levels and goals support a structured presentation of fundamental and advanced test-related concepts to the reader. In this context, the interrelationships between theoretical, technical, and managerial concepts become more apparent. In addition, relationships between the testing process, maturity goals, and such key players as managers, testers and client groups are introduced. Topics and features: - Process/engineering-oriented text - Promotes the growth and value of software testing as a profession - Introduces both technical and managerial aspects of testing in a clear and precise style - Uses the TMM framework to introduce testing concepts in a systematic, evolutionary way to facilitate understanding - Describes the role of testing tools and measurements, and how to integrate them into the testing process Graduate students and industry professionals will benefit from the book, which is designed for a graduate course in software testing, software quality assurance, or software validation and verification Moreover, the number of universities with graduate courses that cover this material will grow, given the evolution in software development as an engineering discipline and the creation of degree programs in software engineering.

This is the digital version of the printed book (Copyright © 1996). Written in a remarkably clear style, *Creating a Software Engineering Culture* presents a comprehensive approach to improving the quality and effectiveness of the software development process. In twenty chapters spread over six parts, Wiegers promotes the tactical changes required to support process improvement and high-quality software development. Throughout the text, Wiegers identifies scores of culture builders and culture killers, and he offers a wealth of references to resources for the software engineer, including seminars, conferences, publications, videos, and on-line information. With case studies on process improvement and software metrics programs and an entire part on action planning (called "What to Do on Monday"), this practical book guides the reader in applying the concepts to real life. Topics include software culture concepts, team behaviors, the five dimensions of a software project, recognizing achievements, optimizing customer involvement, the project champion model, tools for sharing the vision, requirements traceability matrices, the capability maturity model, action planning, testing, inspections, metrics-based project estimation, the cost of quality, and much more! Principles from Part 1 Never let your boss or your customer talk you into doing a bad job. People need to feel the work they do is appreciated. Ongoing education is every team member's responsibility. Customer involvement is the

Read PDF Practical Software Metrics For Project Management And Process Improvement

most critical factor in software quality. Your greatest challenge is sharing the vision of the final product with the customer. Continual improvement of your software development process is both possible and essential. Written software development procedures can help build a shared culture of best practices. Quality is the top priority; long-term productivity is a natural consequence of high quality. Strive to have a peer, rather than a customer, find a defect. A key to software quality is to iterate many times on all development steps except coding: Do this once. Managing bug reports and change requests is essential to controlling quality and maintenance. If you measure what you do, you can learn to do it better. You can't change everything at once. Identify those changes that will yield the greatest benefits, and begin to implement them next Monday. Do what makes sense; don't resort to dogma.

On behalf of the PROFES organizing committee we would like to welcome you to the 4th International Conference on Product Focused Software Process Improvement (PROFES 2002) in Rovaniemi, Finland. The conference was held on the Arctic Circle in exotic Lapland under the Northern Lights just before Christmas time, when Kaamos (the polar night is known in Finnish as "Kaamos") shows its best characteristics. PROFES has established itself as one of the recognized international process improvement conferences. Despite the current economic downturn, PROFES has attracted a record number of submissions. A total of 70 full papers were submitted and the program committee had a difficult task in selecting the best papers to be presented at the conference. The main theme of PROFES is professional software process improvement (SPI) motivated by product and service quality needs. SPI is facilitated by software process assessment, software measurement, process modeling, and technology transfer. It has become a practical tool for quality software engineering and management. The conference addresses both the solutions found in practice and the relevant research results from academia.

Practical Software Metrics for Project Management and Process Improvement

Software Development Patterns and Antipatterns

Applying Software Metrics

Building Reliable Systems

Towards a Synergistic Combination of Research and Practice in Software Engineering

Effectively Meeting Evolving Business Needs

Five Core Metrics

This book is a comprehensive, step-by-step guide to software engineering. This book provides an introduction to software engineering for students in undergraduate and post graduate programs in computers.

This book discusses business architecture as a basis for aligning efforts with outcomes. It views BA as complementary to enterprise architecture, where the focus of technological

Read PDF Practical Software Metrics For Project Management And Process Improvement

initiatives and inventories is to understand and improve business organization, business direction, and business decision-making. This book provides a practical, long-term view on BA. Based on the authors' consulting experience and industrial research, the material in this book is a valuable addition to the thought processes around BA and EA. The lead author has direct and practical experience with large clients in applying APQC capability framework for undertaking multiple enterprise-wide capability assessments.

Features a useful collection of important and practical papers on applying software metrics and measurement. The book details the importance of planning a successful measurement program with a complete discussion of why, what, where, when, and how to measure and who should be involved. Each chapter addresses these significant questions and provides the essential answers in building an effective measurement program. The book differs from others on the market by focusing on the application of the metrics rather than the metrics themselves. The author's provide information based on actual experience with successful metrics programs. Each chapter includes a case study focusing on technology transfer and a set of recommended references. The book serves as a guide on the use and application of software metrics in industrial environments. It is specially designed for managers, product supervisors, and quality assurance personnel who want to know how to implement a metrics program.

A comprehensive reference manual to the Certified Software Quality Engineer Body of Knowledge and study guide for the CSQE exam.

Synergizing Strategies and Intelligence with Architecture

Current Trends in Research and Practice

Metrics-driven Enterprise Software Development

Software Best Practice 4

Practical Software Metrics for Project Management and Process Improvement

Successful Software Development

Software Process and Product Measurement

Practical approach to software measurement Contains hands-on industry experiences

C. Amting Directorate General Information Society, European Commission, Brussels Under the 4th Framework of European Research, the European Systems and Software Initiative (ESSI) was part of the ESPRIT Programme. This initiative funded more than 470 projects in the area' of software and system process improvements. The majority of these projects were process improvement experiments carrying out and taking

up new development processes, methods and technology within the software development process of a company. In addition, nodes (centres of expertise), European networks (organisations managing local activities), training and dissemination actions complemented the process improvement experiments. ESSI aimed at improving the software development capabilities of European enterprises. It focused on best practice and helped European companies to develop world class skills and associated technologies to build the increasingly complex and varied systems needed to compete in the marketplace. The dissemination activities were designed to build a forum, at European level, to exchange information and knowledge gained within process improvement experiments. Their major objective was to spread the message and the results of experiments to a wider audience, through a variety of different channels. The European Experience Exchange (URX) project has been one of these dissemination activities within the European Systems and Software Initiative. URX has collected the results of practitioner reports from numerous workshops in Europe and presents, in this series of books, the results of Best Practice achievements in European Companies over the last few years.

IT Innovation for Adaptability and Competitiveness addresses the topic of IT innovations that can further an organization's ability to adapt and be competitive. Thus we address the problem at an earlier starting point, that is, the emergence of something innovative in an organization, applied to that organization, and its process of being diffused and accepted internally. Topics covered in the book include: -The role of IT in organizational innovation, -Innovating systems development & process, -Assessing innovation drivers, -Innovation adoption, -New environments, new innovation practices. This volume contains the edited proceedings of the Seventh Working Conference on IT Innovation for Adaptability and Competitiveness, which was sponsored by the International Federation for Information Processing (IFIP) Working Group 8.6 and held at Intel Corporation, Leixlip, Ireland in May-June 2004.

Based on CMM^o/CMMI^o, this unique new resource offers you practical "how to" guidance on software process improvement. The book provides you with clear implementation steps that are designed to have a highly positive impact on even your most challenging projects. You get valuable, time-saving

templates that can be quickly tailored for your purposes and used immediately. The book reveals the most common problem areas that effect projects and processes, and offers specific advice on how to overcome these pitfalls. You learn how to assess the strengths and weaknesses of your organization and your specific projects, and discover the most effective ways to address the critical areas that need attention.

Product Focused Software Process Improvement

A Process-Oriented Approach

CMM in Practice

Software Process Improvement: Metrics, Measurement, and Process Modelling

International Conferences IWSM 2008, Metrikon 2008, and Mensura 2008 Munich, Germany, November 18-19, 2008.

Proceedings

A Measurement Framework for Software Projects

Software Development

Software Engineering for Image Processing Systems creates a modern engineering framework for the specification, design, coding, testing, and maintenance of image processing software and systems. The text is designed to benefit not only software engineers, but also workers with backgrounds in mathematics, the physical sciences, and other engineering

You CAN Turn Around A Failing Project! Poor project results are all too common and result in dissatisfied customers, users, and project staff. With countless people, goals, objectives, expectations, budgets, schedules, deliverables, and deadlines to consider, it can be difficult to keep projects in focus and on track. How to Save a Failing Project: Chaos to Control arms project managers with the tools and techniques needed to address these project challenges. The authors provide guidance to develop a project plan, establish a schedule for execution, identify project tracking mechanisms, and implement turnaround methods to avoid failure and regain control. With this valuable resource you will be able to:

- Identify key factors leading to failure*
- Learn how to recover a failing project and minimize future risk*
- Better analyze your project by defining proper business objectives and goals*
- Gain insight on industry best practices for planning*

Discusses how to define and organize use cases that model the user requirements of a software application. The approach focuses on identifying all the parties who will be using the system, then writing detailed use case descriptions and structuring the use case model. An ATM example runs throughout the book. The authors work at Rational Software. Annotation copyrighted by Book News, Inc., Portland, OR

Access Versus Ownership to Word Formation in Language and Computation

Global and Industry Perspectives

Software Project Management in Practice

Practical Software Process Improvement

IFIP TC8/WG8.6 Seventh Working Conference on IT Innovation for Adaptability and Competitiveness May 30-June 2, 2004, Leixlip, Ireland

IFIP TC5 WG5.4 3rd International Conference on Reliability, Quality and Safety of Software-Intensive Systems (ENCRESS '97), 29th-30th May 1997, Athens, Greece

Software Measurement

Chaos to Control

Read PDF Practical Software Metrics For Project Management And Process Improvement

"A clearly written book that is a useful primer for a very complicated set of topics."

--Capers Jones, Chief Scientist Emeritus, Software Productivity Research LLC

Practical Software Estimation brings together today's most valuable tips, techniques, and best practices for accurately estimating software project efforts, costs, and schedules. Written by a leading expert in the field, it addresses the full spectrum of real-world challenges faced by those who must develop reliable estimates. M. A. Parthasarathy draws on the immense experience of Infosys, one of the world's largest and most respected providers of IT-enabled business solutions, to bring you the only book with detailed guidance on estimating insourced and outsourced software projects, as well as projects that blend both approaches. He demonstrates how to successfully utilize Function Point (FP) methods, the industry's leading estimation model. Then, using real case studies, he systematically identifies pitfalls that can lead to inaccurate estimates--and offers proven solutions. Coverage includes

- How to estimate all types of software projects, including "fresh" development, reengineering, and maintenance
- How to incorporate the impact of core project elements on estimates: scope, environment, experience, and tools
- FP analysis from start to finish: data and transaction functions, general system characteristics, and more FP methods for any platform or business function
- Innovative re-estimation methods to track progress
- How to quote RFPs and prepare contracts: fixed price, time/material, and project execution lifecycle models
- Alternatives to FP: Delphi, COCOMO II, and COSMIC-FFP
- How to choose the right estimation tools

Practical Software Estimation is the definitive reference for anyone who must estimate software projects accurately: project and IT managers, individual developers, system designers, architects, executives, consultants, and outsourcers alike.

List of Figures
List of Tables
Foreword
Preface
Acknowledgments
Chapter 1: Introduction
Chapter 2: Role of Estimation in Software Projects
Chapter 3: A Study of Function Point Analysis
Chapter 4: Data Functions
Chapter 5: Transactional Functions
Chapter 6: General System Characteristics
Chapter 7: Size, Effort, and Scheduling of Projects
Chapter 8: Estimation Flavors
Chapter 9: A Sense of Where You Are
Chapter 10: Tips, Tricks, and Traps
Chapter 11: Insourcing versus Outsourcing
Chapter 12: Key Factors in Software Contracts
Chapter 13: Project Estimation and Costing
Chapter 14: Other Estimation Methods
Chapter 15: Estimation Tools
Chapter 16: Estimation Case Study
Appendix A: Reference Tables: Transaction Function Counts
Appendix B: Reference Tables: Data Function Points
Bibliography
Index

PART I: FUNDAMENTALS OF MEASUREMENT AND EXPERIMENTATION

1. Measurement: What Is It and Why Do It?
2. The Basics of Measurement
3. A Goal-Based Framework for Software Measurement
4. Empirical Investigation
5. Software Metrics Data Collection
6. Analyzing Software-Measurement Data

PART II: SOFTWARE-ENGINEERING MEASUREMENT

7. Measuring Internal Product Attributes: Size
8. Measuring Internal Product Attributes: Structure
9. Measuring Internal Product Attributes
10. Software Reliability: Measurement and Prediction
11. Resource Measurement: Productivity, Teams, and Tools
12. Making Process Predictions

PART III: MEASUREMENT AND MANAGEMENT

13. Planning a Measurement Program
14. Measurement in Practice
15. Empirical Research in Software Engineering

APPENDIXES:

- A. Solutions to Selected Exercises
- B. Metric Tools
- C. Acronyms and Glossary

ANNOTATED BIBLIOGRAPHY / INDEX

The object oriented paradigm has become one of the dominant forces in the computing

Read PDF Practical Software Metrics For Project Management And Process Improvement

world. According to a recent survey, by the year 2000, more than 80% of development organizations are expected to use object technology as the basis for their distributed development strategies. Handbook of Object Technology encompasses the entire spectrum of disciplines and topics related to this rapidly expanding field - outlining emerging technologies, latest advances, current trends, new specifications, and ongoing research. The handbook divides into 13 sections, each containing chapters related to that specific discipline. Up-to-date, non-abstract information provides the reader with practical, useful knowledge - directly applicable to the understanding and improvement of the reader's job or the area of interest related to this technology. Handbook of Object Technology discusses: the processes, notation, and tools for classical OO methodologies as well as information on future methodologies prevalent and emerging OO languages standards and specifications frameworks and patterns databases metrics business objects intranets analysis/design tools client/server application development environments

Software projects today are often characterized by poor quality, schedule overruns and high costs. Hence project decision makers need an objective and validated measurement framework to allocate limited resources and to track project progress. In this backdrop, based on the Goal-Question-Metric (GQM) model, Prashanth Harish Southeikal has come up with eight generic objective measures for the project stakeholders to base their corrective actions for successful project delivery . The measurement framework is validated (i) theoretically with measurement theory criteria and (ii) empirically with case studies (Controlled and Uncontrolled) including a global survey representing industry practitioners from 29 countries.

Practical Software Testing

Formalized Approaches

IT Innovation for Adaptability and Competitiveness

Software Engineering for Image Processing Systems

Function Point Methods for Insourced and Outsourced Projects

4th International Conference, PROFES 2002 Rovaniemi, Finland, December 9-11, 2002, Proceedings

Encyclopedia of Library and Information Science

This is the digital version of the printed book (Copyright © 2003). To succeed in the software industry, managers need to cultivate a reliable development process. By measuring what teams have achieved on previous projects, managers 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 managers 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, managers can adjust ongoing projects to changing conditions—surprises that would otherwise cause project failure.

Drawing on best practices identified at the Software Quality Institute and embodied in bodies of knowledge from the Project Management Institute, the

American Society of Quality, IEEE, and the Software Engineering Institute, Quality Software Project Management teaches 34 critical skills that allow any manager to minimize costs, risks, and time-to-market. Written by leading practitioners Robert T. Futrell, Donald F. Shafer, and Linda I. Shafer, it addresses the entire project lifecycle, covering process, project, and people. It contains extensive practical resources-including downloadable checklists, templates, and forms.

Software Development is the most thorough, realistic guide to "what works" in software development - and how to make it happen in your organization. Leading consultant Marc Hamilton tackles all three key elements of successful development: people, processes, and technology. From streamlining infrastructures to retraining programmers, choosing tools to implementing service level agreements, Hamilton unifies all of today's best practices - in management, architecture, and software engineering.

For over 20 years, Software Engineering: A Practitioner's Approach has been the best selling guide to software engineering for students and industry professionals alike. The sixth edition continues to lead the way in software engineering. A new Part 4 on Web Engineering presents a complete engineering approach for the analysis, design, and testing of Web Applications, increasingly important for today's students. Additionally, the UML coverage has been enhanced and significantly increased in this new edition. The pedagogy has also been improved in the new edition to include sidebars. They provide information on relevant software tools, specific work flow for specific kinds of projects, and additional information on various topics. Additionally, Pressman provides a running case study called "Safe Home" throughout the book, which provides the application of software engineering to an industry project. New additions to the book also include chapters on the Agile Process Models, Requirements Engineering, and Design Engineering. The book has been completely updated and contains hundreds of new references to software tools that address all important topics in the book. The ancillary material for the book includes an expansion of the case study, which illustrates it with UML diagrams. The On-Line Learning Center includes resources for both instructors and students such as checklists, 700 categorized web references, Powerpoints, a test bank, and a software engineering library-containing over 500 software engineering papers. TAKEAWY HERE IS THE FOLLOWING:1. AGILE PROCESS METHODS ARE COVERED EARLY IN CH. 42. NEW PART ON WEB APPLICATIONS --5 CHAPTERS

Practical Software Estimation

Objective Information for Decision Makers

The Unified Process Inception Phase

Reliability, Quality and Safety of Software-Intensive Systems

The Certified Software Quality Engineer Handbook

Outcome-Driven Business Architecture

A Practical Metrics and Measurements Guide for Today's Software Project Manager

Your Complete Guide to Project Management Metrics is Here! Metrics for Project Management: A Formalized Approach describes a comprehensive set of project management metrics in an easy-to-read format. Through a unique presentation of metrics through the categories of "things," "people," and "enterprise," you'll learn how metrics can:

- Guide you toward informed decisions
- Help the enterprise recognize the sum of its collective capabilities
- Ensure that plans for producing and delivering products and services are consistently realistic, achievable, and attainable
- Link the efforts of individual team members with the overall success of the project
- Indirectly promote teamwork and improve team morale

Software development has been a troubling since it first started. There are seven chronic problems that have plagued it from the beginning: Incomplete and ambiguous user requirements that grow by >2% per month. Major cost and schedule overruns for large applications > 35% higher than planned. Low defect removal efficiency (DRE) Cancelled projects that are not completed: > 30% above 10,000 function points. Poor quality and low reliability after the software is delivered: > 5 bugs per FP. Breach of contract litigation against software outsource vendors. Expensive maintenance and enhancement costs after delivery. These are endemic problems for software executives, software engineers and software customers but they are not insurmountable. In Software Development Patterns and Antipatterns, software engineering and metrics pioneer Capers Jones presents technical solutions for all seven. The solutions involve moving from harmful patterns of software development to effective patterns of software development. The first section of the book examines common software development problems that have been observed in many companies and government agencies. The data on the problems comes from consulting studies, breach of contract lawsuits, and the literature on major software failures. This section considers the factors involved with cost overruns, schedule delays, canceled projects, poor quality, and expensive maintenance after deployment. The second section shows patterns that lead to software success. The data comes from actual companies. The section's first chapter on Corporate Software Risk Reduction in a Fortune 500 company was based on a major telecom company whose CEO was troubled by repeated software failures. The other chapters in this section deal with methods of achieving excellence, as well as measures that can prove excellence to C-level executives, and with continuing excellence through the maintenance cycle as well as for software development.

Project initiation; Project planning; Project execution and termination. Is the Unified Process the be all and end all standard for developing object-oriented component-based software? This book is the third in a four volume series that presents a critical review of the Unified Process. The authors present a survey of the alte

Handbook of Object Technology

Volume 64 - Supplement 27 - Access Versus Ownership to Word Formation in Language and Computation

*A Generic and Practical Goal-Question-Metric(Gqm) Based Approach.
Creating a Software Engineering Culture
Processes for Executing Software Projects at Infosys
Quantifying Software
Software Metrics*

Annotation *This book constitutes the thoroughly refereed post-proceedings of the International Workshop on Software Measurement, IWSM-Mensura 2007, held in Palma de Mallorca, Spain, in November 2007. The 16 revised full papers presented were carefully reviewed and selected for inclusion in the book. The papers deal with aspects of software measurement like function-points measurement, effort and cost estimates, prediction, industrial experiences in software measurement, planning and implementing measurement, measurement-based software process improvement, best practices in software measurement, usability and user interaction measurement, measurement of open source projects, teaching and learning software measurement as well as new trends and ontologies for software measurement.*

Software developers are faced with the challenge of making software systems and products of ever greater quality and safety, while at the same time being faced with the growing pressure of costs reduction in order to gain and maintain competitive advantages. As in any scientific and engineering discipline, reliable measurement is essential for talking on such a challenge. "Software measurement is an excellent abstraction mechanism for learning what works and what doesn't" (Victor Basili). Measurement of both software process and products provides a large amount of basic information for the evaluation of the software development processes or the software products themselves. Examples of recent successes in software measurement span multiple areas, such as evaluation of new development methods and paradigms, quality and management improvement programs, tool-supporting initiatives and company wide measurement programs. The German Computer Science Interest (GI) Group of Software Metrics and the Canadian Interest Group in Software Metrics (CIM) have attended to these concerns in the recent years. Research initiatives were directed initially to the definition of software metrics and then to validation of the software metrics themselves. This was followed by more and more investigation into practical applications of software metrics and by critical analysis of the benefits and weaknesses of software measurement programs. Key findings in this area of software engineering have been published in some important books, such as Dumke and Zuse's Theory and Practice of Software Measurement, Ebert and Dumke's Software Metrics in Practice and Lehner, Dumke and Abran's Software Metrics.

Rev. ed. of: *Cultivating successful software development. c1997.*

This book reports on recent advances in software engineering research and practice. Divided into 15 chapters, it addresses: languages and tools; development processes; modelling, simulation and verification; and education. In the first category, the book includes chapters on domain-specific languages, software complexity, testing and tools. In the second, it reports on test-driven development, processing of business rules, and software management. In turn, subsequent chapters address modelling, simulation and verification of real-time systems, mobile systems and computer networks, and a scrum-based framework. The book was written by researchers and practitioners, the goal being to achieve a synergistic combination of research results achieved in academia and best practices used in the industry, and to provide a valuable reference guide for both groups.

How to Save a Failing Project

Metrics for Project Management

A Practitioner's Approach

Software Engineering

Best Practices in Implementing the UP

Agile Approaches for Successfully Managing and Executing Projects in the Fourth Industrial Revolution

Practical Software Measurement

Part of the new Digital Filmmaker Series! Digital Filmmaking: An Introduction is the first book in the new Digital Filmmaker Series. Designed for an introductory level course in digital filmmaking, it is intended for anyone who has an interest in telling stories with pictures and sound and won't assume any familiarity with equipment or concepts on the part of the student. In addition to the basics of shooting and editing, different story forms are introduced from documentary and live events through fictional narratives. Each of the topics is covered in enough depth to allow anyone with a camera and a computer to begin creating visual projects of quality.

Communication between man and machine is vital to completing projects in the current day and age. Without this constant connectiveness as we enter an era of big data, project completion will result in utter failure. Agile Approaches for Successfully Managing and Executing Projects in the Fourth Industrial Revolution addresses changes wrought by Industry 4.0 and its effects on project management as well as adaptations and adjustments that will need to be made within project life cycles and project risk management. Highlighting such topics as agile planning, cloud projects, and organization structure, it is designed for project managers, executive management, students, and academicians.

It is, indeed, widely acceptable today that nowhere is it more important to focus on the improvement of software quality than in the case of systems with requirements in the areas of safety and reliability - especially for distributed,

real-time and embedded systems. Thus, much research work is under progress in these fields, since software process improvement impinges directly on achieved levels of quality, and many application experiments aim to show quantitative results demonstrating the efficacy of particular approaches. Requirements for safety and reliability - like other so-called non-functional requirements for computer-based systems - are often stated in imprecise and ambiguous terms, or not at all. Specifications focus on functional and technical aspects, with issues like safety covered only implicitly, or not addressed directly because they are felt to be obvious; unfortunately what is obvious to an end user or system user is progressively less so to others, to the extent that a software developer may not even be aware that safety is an issue. Therefore, there is a growing evidence for encouraging greater understanding of safety and reliability requirements issues, right across the spectrum from end user to software developer; not just in traditional safety-critical areas (e.g. nuclear, aerospace) but also acknowledging the need for such things as heart pacemakers and other medical and robotic systems to be highly dependable.

Metrics for software development are usually employed ad-hoc and without clear directions for interpreting the numbers and acting on them. Almost every other engineering discipline has clear guidelines for measuring processes and products and making decisions based on quantified evidence. This practical book describes how to integrate processes and metrics to ensure easier and more effective enterprise software development. It crosses the divide between theory and practice and also discusses why essential processes so often fail to deliver quality industrial software. Enterprise Software Development introduces the techniques for building, applying and interpreting metrics for the workflows across the software development life cycle phases of inception, elaboration, construction and transition. It is a must read for software engineering practitioners (architects, application developers, designers and project managers), academics, and students and apprentices of software engineering.

Use Case Modeling

Best Practices in Software Measurement

A Rigorous and Practical Approach

Managing Systems and IT Projects

How to use metrics to improve project and process performance

Quality Software Project Management

The Intelligence Behind Successful Software Management

Software is one of the most important products in human history and is widely used by all industries and all countries. It is also one of the most expensive and labor-intensive products in human history. Software also has very poor quality that has caused many major disasters and wasted many millions of dollars.

Read PDF Practical Software Metrics For Project Management And Process Improvement

Software is also the target of frequent and increasingly serious cyber-attacks. Among the reasons for these software problems is a chronic lack of reliable quantified data. This reference provides quantified data from many countries and many industries based on about 26,000 projects developed using a variety of methodologies and team experience levels. The data has been gathered between 1970 and 2017, so interesting historical trends are available. Since current average software productivity and quality results are suboptimal, this book focuses on "best in class" results and shows not only quantified quality and productivity data from best-in-class organizations, but also the technology stacks used to achieve best-in-class results. The overall goal of this book is to encourage the adoption of best-in-class software metrics and best-in-class technology stacks. It does so by providing current data on average software schedules, effort, costs, and quality for several industries and countries. Because productivity and quality vary by technology and size, the book presents quantitative results for applications between 100 function points and 100,000 function points. It shows quality results using defect potential and DRE metrics because the number one cost driver for software is finding and fixing bugs. The book presents data on cost of quality for software projects and discusses technical debt, but that metric is not standardized. Finally, the book includes some data on three years of software maintenance and enhancements as well as some data on total cost of ownership. This application-oriented book shows how to apply proven software metrics and methods to efficiently manage software development and maintenance--to help boost productivity, efficiency, and quality of software projects at every stage of the process. Detailing practical methods throughout, the book covers tips to best measure and present progress, a useful model for understanding organization limitations, possible problems in process improvement illustrated by examples, evidence of what works and what doesn't work, and more. An ideal reference for project managers and professionals responsible for process improvement.