Metrics and Models in Software Quality Engineering (paperback) (2nd Edition) By Stephen H. Kan

"This is the single best book on software quality engineering and metrics that I've encountered."--Capers Jones, from the Foreword Metrics and Models in Software Quality Engineering, Second Edition, is the definitive book on this essential topic of software development. Comprehensive in scope with extensive industry examples, it shows how to measure software quality and use measurements to improve the software development process. Four major categories of quality metrics and models are addressed: quality management, software reliability and projection, complexity, and customer view. In addition, the book discusses the fundamentals of measurement theory, specific quality metrics and tools, and methods for applying metrics to the software development process. New chapters bring coverage of critical topics, including: In-process metrics for software testing Metrics for object-oriented software development Availability metrics Methods for conducting in-process quality assessments and software project assessments Dos and Don'ts of Software Process Improvement, by Patrick O'Toole Using Function Point Metrics to Measure Software Process Improvement, by Capers Jones In addition to the excellent balance of theory, techniques, and examples, this book is highly instructive and practical, covering one of the most important topics in software development--quality engineering.

0201729156B08282002

Download Metrics and Models in Software Quality Engineering ...pdf

Read Online Metrics and Models in Software Quality Engineeringi ...pdf
Metrics and Models in Software Quality Engineering (paperback) (2nd Edition) By Stephen H. Kan

Metrics and Models in Software Quality Engineering (paperback) (2nd Edition) By Stephen H. Kan

"This is the single best book on software quality engineering and metrics that I've encountered."--Capers Jones, from the Foreword.

Metrics and Models in Software Quality Engineering, Second Edition, is the definitive book on this essential topic of software development. Comprehensive in scope with extensive industry examples, it shows how to measure software quality and use measurements to improve the software development process. Four major categories of quality metrics and models are addressed: quality management, software reliability and projection, complexity, and customer view. In addition, the book discusses the fundamentals of measurement theory, specific quality metrics and tools, and methods for applying metrics to the software development process. New chapters bring coverage of critical topics, including: In-process metrics for software testing, Metrics for object-oriented software development, Availability metrics, Methods for conducting in-process quality assessments and software project assessments, Dos and Don'ts of Software Process Improvement, by Patrick O'Toole, Using Function Point Metrics to Measure Software Process Improvement, by Capers Jones. In addition to the excellent balance of theory, techniques, and examples, this book is highly instructive and practical, covering one of the most important topics in software development--quality engineering. 020172915608282002

Metrics and Models in Software Quality Engineering (paperback) (2nd Edition) By Stephen H. Kan

Bibliography

- Sales Rank: #2047526 in Books
- Published on: 2002-09-26
- Original language: English
- Number of items: 1
- Dimensions: 9.00" h x 1.20" w x 7.40" l, .0 pounds
- Binding: Paperback
- 560 pages

Download Metrics and Models in Software Quality Engineering ...pdf

Read Online Metrics and Models in Software Quality Engineeri ...pdf
Editorial Review

From the Back Cover

"This is the single best book on software quality engineering and metrics that I've encountered."
--Capers Jones, from the Foreword

**Metrics and Models in Software Quality Engineering, Second Edition**, is the definitive book on this essential topic of software development. Comprehensive in scope with extensive industry examples, it shows how to measure software quality and use measurements to improve the software development process. Four major categories of quality metrics and models are addressed: quality management, software reliability and projection, complexity, and customer view. In addition, the book discusses the fundamentals of measurement theory, specific quality metrics and tools, and methods for applying metrics to the software development process.

New chapters bring coverage of critical topics, including:

- In-process metrics for software testing
- Metrics for object-oriented software development
- Availability metrics
- Methods for conducting in-process quality assessments and software project assessments
- Dos and Don'ts of Software Process Improvement, by Patrick O'Toole
- Using Function Point Metrics to Measure Software Process Improvement, by Capers Jones

In addition to the excellent balance of theory, techniques, and examples, this book is highly instructive and practical, covering one of the most important topics in software development--quality engineering.

0201729156B08282002

About the Author

**Stephen H. Kan** is Senior Technical Staff Member (STSM) and a technical manager in programming for IBM in Rochester, Minnesota. As process manager of the quality management process in product development for IBM's eServer iSeries software development, his responsibilities include quality goal setting, supplier quality requirements, quality plans, in-process metrics, field quality status, and quality and project assessments. Dr. Kan has been a faculty member of the Master of Science in Software Engineering program at the University of Minnesota since 1998. He is certified by the American Society for Quality as a Quality Engineer, a Reliability Engineer, and a Quality Manager, and by the Project Management Institute as a Project Management Professional.
Looking at software engineering from a historical perspective, the 1960s and earlier could be viewed as the functional era, the 1970s the schedule era, the 1980s the cost era, and the 1990s and beyond the quality and efficiency era. In the 1960s we learned how to exploit information technology to meet institutional needs and began to link software with the daily operations of institutions. In the 1970s, as the industry was characterized by massive schedule delays and cost overruns, the focus was on planning and control of software projects. Phase-based life-cycle models were introduced, and analysis, like the mythical man-month, emerged. In the 1980s hardware costs continued to decline, and information technology permeated every facet of our institutions and became available to individuals. As competition in the industry became keen and low-cost applications became widely implemented, the importance of productivity in software development increased significantly. Various software engineering cost models were developed and used. In the late 1980s, the importance of quality was also recognized.

The 1990s and beyond is certainly the quality era. With state-of-the-art technology now able to provide abundant functionality, customers demand high quality. Demand for quality is further intensified by the ever-increasing dependence of society on software. Billing errors, large-scale disrupted telephone services, and even missile failures during recent wars can all be traced to the issue of software quality. In this era, quality has been brought to the center of the software development process. From the standpoint of software vendors, quality is no longer an advantage factor in the marketplace; it has become a necessary condition if a company is to compete successfully.

Starting in the mid 1990s two major factors emerged that have proved to have an unprecedented impact on not only software engineering but also on global business environments: business reengineering for efficiency and the Internet. Software development has to be more efficient and the quality level of the delivered products has to be high to meet requirements and to be successful. This is especially the case for mission-critical applications. The adverse impact of poor quality is much more significant and at a much wider scale; the quality "dikes" that software is supposed to provide are never more important. These factors will continue to affect software engineering for many years to come during this new millennium.

Measurement plays a critical role in effective and efficient software development, as well as provides the scientific basis for software engineering that makes it a true engineering discipline. This book describes the software quality engineering metrics and models: quality planning, process improvement and quality control, in-process quality management, product engineering (design and code complexity), reliability estimation and projection, and analysis of customer satisfaction data. Many measurement books take an encyclopedic approach, in which every possible software measurement is included, but this book confines its scope to the metrics and models of software quality. Areas such as cost estimation, productivity, staffing, and performance measurement, for which numerous publications exist, are not covered.

In this edition, seven new chapters have been added, covering in-process metrics for software testing, object-oriented metrics, availability metrics, in-process quality assessment, software project assessment, process improvement dos and don'ts, and measuring software process improvement. The chapter that described the AS/400 software quality management system has been eliminated. Updates and revisions have been made throughout the original chapters, and new sections, figures, and tables have been added.
Two of the new chapters are special contributions from two experts. This is a key feature of the new edition. The chapter on the dos and don'ts of software process improvement is contributed by Patrick O'Toole. A highly regarded process improvement expert and with over 20 years of experience, Patrick brings to this book a perspective on process improvement that I share as a practitioner. That perspective is based on practical experience, is project-centric, and is aligned with the strategic business imperative of the organization. Patrick also brings humor to this otherwise serious subject, making the reading of the chapter so enjoyable. The chapter on measuring software process improvement is a special contribution by Capers Jones. A pioneer in software metrics, productivity research, software quality control, and software assessments, Capers's work is well known nationally and internationally. His data-based and fact-based approach in software assessments and benchmarking studies is unparalleled. Based on experience and data from more than 10,000 projects, he brings to the readers a practical approach to software process improvement and the major quantitative findings related to software process improvement. The value of function point metrics is demonstrated via the analyses and findings. The chapter is a must read for software process professionals who are interested in measuring software process improvement.

Another new feature in this edition is a set of recommendations for small teams and organizations that are starting to implement a metrics program, with minimum resources. These recommendations are shown in the form of box inserts in nine of the chapters. A number of examples in the book are based on small team projects, and many methods and techniques are appropriate for large projects as well as small ones. This set of recommendations is from the perspective of small organizations or teams using a small number of metrics, with the intent to effect improvement in their software development effort.

This book is intended for use by software quality professionals; software project managers; software product managers; software development managers; software engineers; software product assurance personnel; and students in software engineering, management information systems, systems engineering, and quality engineering and management. For teachers, it is intended to provide a basis for a course at the upper-division undergraduate or graduate level. A number of software engineering, computer science, and quality engineering programs in the United States and overseas have used the first edition of this book as a text.

Themes of This Book

This book has several themes. First, balancing theory, techniques, and real-life examples, it provides practical guidelines in the practice of quality engineering in software development. Although equations and formulas are involved, the focus is on the understanding and applications of the metrics and models rather than mathematical derivations. Throughout the book, numerous real-life examples are used from the software development laboratory at IBM Rochester, Minnesota, home of the AS/400 and the IBM eServer iSeries computer systems, and from other companies in the software industry. IBM Rochester won the Malcolm Baldrige National Quality Award in 1990. A number of metrics described in this book were being used at that time, and many have been developed and refined since then. All metrics are substantiated by ample implementation experience. IBM Rochester develops and delivers numerous projects of different sizes and types every year, including very large and complex as well as small ones; and they range from firmware, to operating systems, to middleware, to applications.

Second, I attempt to provide a good coverage of the various types of metrics and models in the emerging field of software quality engineering. In addition to general discussions about metrics and techniques, this book categorizes and covers four types of metrics and models: (1) quality management models; (2) software reliability and projection models; (3) complexity metrics and models; and (4) customer-view metrics,
measurements, and models. These metrics and models cover the entire software development process from high-level design to testing and maintenance, as well as all phases of reliability. Furthermore, although this book is not on total quality management (TQM), it is a major consideration in the coverage of metrics. The philosophy of TQM is the linking of product quality and customer satisfaction for the purpose of achieving long-term success. TQM is the reason for including two chapters on customer-view metrics and measurements--availability metrics and customer satisfaction--in addition to the many chapters on product and process metrics. In other discussions in the book, the customer's perspective is included where appropriate.

Third, by linking metrics and models to quality improvement strategies and improvement actions, we attempt to focus on using, not just describing, metrics. A framework for interpreting in-process metrics and assessing in-process quality status--the effort/outcome model--is presented. The direct link between a recommended quality strategy during development and the defect-removal model is shown. Examples of actions tied to specific metrics and analysis are given. Furthermore, to illustrate the metrics, many figures and graphs are used. This is a reflection of the fact that in real-life project and quality management, a clear visual presentation often improves understanding and increases the effectiveness of the metrics.

Fourth, following up on quality and process improvement at a more general level than specific metric discussions, the book continues with chapters that discuss the in-process quality assessment process, a method for conducting software project assessments, practical advice on process improvement dos and don'ts, and quantitative analysis of software process improvement. The common thread underlying these chapters, as with other chapters on metrics and models, is practical experience with industry projects.

Organization of This Book

The following list details the focus of each chapter.

- **Chapter 1, What Is Software Quality?**, discusses the definition of quality and software quality. The customer's role in the definition is highlighted. Quality attributes and their relationships are discussed. The second part of the chapter covers the definition and framework of TQM and the customer's view of quality, a key focus in this book.

- **Chapter 2, Software Development Process Models**, reviews various development process models that are used in the software industry. It briefly describes two methods of software process maturity assessment--the SEI process capability maturity model (CMM) (by the Software Engineering Institute) and the SPR assessment method (by the Software Productivity Research, Inc.). It summarizes two bodies of quality management standards--the Malcolm Baldrige National Quality Award assessment discipline and ISO 9000.

- **Chapter 3, Fundamentals of Measurement Theory**, examines measurement theory fundamentals, which are very important for the practice of software measurement. The concept of operational definition and its importance in measurement are illustrated with an example. The level of measurement, some basic measures, and the concept of six sigma are discussed. The two key criteria of measurement quality, reliability and validity, and the related issue of measurement errors are examined and their importance is articulated. This chapter also provides a discussion on correlation and addresses the criteria necessary to establish causality based on observational data.

- **Chapter 4, Software Quality Metrics Overview**, presents examples of quality metrics for the three categories of metrics associated with the software life cycle: end-product, in-process, and maintenance. It describes the metrics programs of several large software companies and discusses collection of software engineering data.
Chapter 5, Applying the Seven Basic Quality Tools in Software Development, describes the application of the basic statistical tools for quality control, known as Ishikawa's seven basic tools, in software development. The potentials and challenges of applying the control chart in software environments are discussed. In addition, a qualitative tool for brainstorming and for displaying complex cause-and-effect relationships—the relations diagram—is discussed.

Chapter 6, Defect Removal Effectiveness, is the first of five chapters about the models and metrics that describe the quality dynamics of software development. Through two types of models, quality management models and software reliability and projection models, the quality of software development can be planned, engineered, managed, and projected. This chapter examines the central concept of defect removal effectiveness, its measurements, and its role in quality planning.

Chapter 7, The Rayleigh Model, describes the model and its implementation as a reliability and projection model. The Rayleigh Model's use as a quality management model is discussed in Chapter 9.

Chapter 8, Exponential Distribution and Reliability Growth Models, discusses the exponential distribution and the major software reliability growth models. These models, like the Rayleigh Model, are used for quality projection before the software is shipped to customers, just before development is complete. The models are also used for maintenance planning, to model the failure pattern or the defect arrival patterns in the field.

Chapter 9, Quality Management Models, describes several quality management models that cover the entire development cycle. In-process metrics and reports that support the models are shown and discussed. A framework for interpreting in-process metrics and assessing in-process quality status—the effort/outcome model—is presented.

Chapter 10, In-Process Metrics for Software Testing, is a continuation of Chapter 9; it focuses on the metrics for software testing. The effort/outcome model, as it applies to metrics during the testing phase, is elaborated. Candidate metrics for acceptance testing to evaluate vendor-developed software, and the central question of how to know your product is good enough to ship, are also discussed.

Chapter 11, Complexity Metrics and Models, discusses the third type of metrics and models in software engineering. While quality management models and reliability and projection models are for project management and quality management, the objective of the complexity metrics and models is for software engineers to be able to improve their design and implementation of software development.

Chapter 12, Metrics and Lessons Learned for Object-Oriented Projects, covers design and complexity metrics, productivity metrics, quality and quality management metrics for object-oriented development, and lessons learned from the deployment and implementation of OO projects. The first section can be viewed as a continuation of the discussion on complexity metrics and models; the other sections fall within the framework of quality and project management.

Chapter 13, Availability Metrics, discusses system availability and outage metrics, and explores the relationships among availability, reliability, and the traditional defect-rate measurement. Availability metrics and customer satisfaction measurements are the fourth type of metrics and models—customer-oriented metrics.

Chapter 14, Measuring and Analyzing Customer Satisfaction, discusses data collection and measurements of customer satisfaction, and techniques and models for the analysis of customer satisfaction data. From Chapter 3 to this chapter, the entire spectrum of metrics and models is covered.

Chapter 15, Conducting In-Process Quality Assessments, describes in-process quality assessments as an integrated element of good project quality management. Quality assessments are based on both quantitative indicators, such as those discussed in previous chapters, and qualitative information.

Chapter 16, Conducting Software Project Assessments, takes the discussion to yet another level; this chapter proposes a software project assessment method. The focus is at the project level and the discussion is from a practitioner's perspective.

Chapter 17, Dos and Don'ts of Software Process Improvement by Patrick O'Toole, offers practical advice for software process improvement professionals. It provides a link to the process maturity discussions in
Chapter 2.

- Chapter 18, *Using Function Point Metrics to Measure Software Process Improvement* by Capers Jones, discusses the six stages of software process improvement. Based on a large body of empirical data, it examines the costs and effects of process improvement. It shows the results of quantitative analyses with regard to costs, time, schedule, productivity, and quality. It articulates the value of Function Point metrics. It provides a link to the process maturity discussions in Chapter 2.

- Chapter 19, *Concluding Remarks*, provides several observations with regard to software measurement in general and software quality metrics and models in particular, and it offers a perspective on the future of software engineering measurement.

- In the *Appendix*, a real-life example of a project assessment questionnaire is shown. Per the methods and techniques discussed in Chapter 16, readers can customize the questionnaire for their project assessment efforts.

Suggested Ways to Read This Book

The chapters of this book are organized for reading from beginning to end. Later chapters refer to concepts and discussions in earlier chapters. At the same time, each chapter addresses a separate topic and chapters in some groups are more closely coupled than others. Some readers may choose to read specific topics or decide on different starting points. For example, those who are not interested in quality definitions, process models, and measurement fundamentals discussions can start with Chapter 4, Software Quality Metrics Overview. Those who intend to immediately get to the central topics of defect removals, metrics and models for quality planning, and management and projection can start with Chapter 6, Defect Removal Effectiveness. In general, I recommend that the chapters be read in groups, as follows.

- Chapters 1 through 3
  - Chapter 4
  - Chapter 5
- Chapters 6 through 10
- Chapters 11 and 12
- Chapters 13 and 14
- Chapters 15 through 18
- Chapter 19

0201729156P08282002

Users Review

From reader reviews:

Tamera Duckett:

Book is to be different for every grade. Book for children until eventually adult are different content. We all know that that book is very important normally. The book Metrics and Models in Software Quality Engineering (paperback) (2nd Edition) ended up being making you to know about other know-how and of course you can take more information. It is quite advantages for you. The reserve Metrics and Models in Software Quality Engineering (paperback) (2nd Edition) is not only giving you much more new information but also to become your friend when you truly feel bored. You can spend your own personal spend time to read your reserve. Try to make relationship together with the book Metrics and Models in Software Quality

Cindy Martin:

As people who live in often the modest era should be upgrade about what going on or facts even knowledge to make them keep up with the era which is always change and make progress. Some of you maybe will probably update themselves by looking at books. It is a good choice for you but the problems coming to anyone is you don't know what kind you should start with. This Metrics and Models in Software Quality Engineering (paperback) (2nd Edition) is our recommendation to cause you to keep up with the world. Why, since this book serves what you want and want in this era.

Derrick Minor:

Reading a book can be one of a lot of task that everyone in the world likes. Do you like reading book so. There are a lot of reasons why people enjoy it. First reading a book will give you a lot of new facts. When you read a publication you will get new information due to the fact book is one of many ways to share the information or their idea. Second, reading through a book will make you more imaginative. When you studying a book especially fictional works book the author will bring someone to imagine the story how the characters do it anything. Third, you may share your knowledge to some others. When you read this Metrics and Models in Software Quality Engineering (paperback) (2nd Edition), you can tells your family, friends in addition to soon about yours book. Your knowledge can inspire the mediocre, make them reading a publication.

Robin Adams:

In this era which is the greater individual or who has ability in doing something more are more special than other. Do you want to become one of it? It is just simple solution to have that. What you must do is just spending your time almost no but quite enough to enjoy a look at some books. Among the books in the top checklist in your reading list is Metrics and Models in Software Quality Engineering (paperback) (2nd Edition). This book which is qualified as The Hungry Hillsides can get you closer in getting precious person. By looking way up and review this guide you can get many advantages.

Download and Read Online Metrics and Models in Software Quality Engineering (paperback) (2nd Edition) By Stephen H. Kan #TAOFB3LZK8G