

IEEE Standard for a Software Quality Metrics Methodology

Sponsor

Software Engineering Standards Committee
of the
IEEE Computer Society

Reaffirmed 24 June 2004
Approved 8 December 1998

IEEE-SA Standards Board

Reaffirmed 21 January 2005
Approved 16 November 1999

American National Standards Institute

Abstract: A methodology for establishing quality requirements and identifying, implementing, analyzing and validating the process and product software quality metrics is defined. The methodology spans the entire software life cycle.

Keywords: direct metric, metrics framework, quality factor, quality subfactor, software quality metric

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2005 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1995. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Incorporated.

ISBN 1-55937-529-9

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (978) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

(This introduction is not part of IEEE Std 1061-1998, IEEE Standard for a Software Quality Metrics Methodology.)

History

In February 1984, a project to develop a standard for a software quality metrics methodology was approved, and a working group was formed, because there was no existing IEEE standard covering the field of software quality metrics. In December 1992, the IEEE Standards Board approved IEEE Std 1061-1992. It was published by IEEE on 12 March, 1993. This was the first IEEE standard that dealt with quality metrics.

It is important that users of this standard understand that this is a process standard, and not a standard that mandates specific metrics for use. The philosophy of this standard is that an organization can employ whichever metrics it deems most appropriate for its applications, as long as the methodology is followed and the metrics are validated. Another reason for this approach is that there was no consensus on which metrics to mandate for use (the provisions of a standard are mandatory, not optional). Consistent with this approach was the Working Group charter, as provided in the IEEE Standards Board approval of the project authorization request (PAR), which called for a standard methodology to be developed.

Due to the IEEE rule that a standard must be revised or reaffirmed within five years of issuance, a PAR for a revision was submitted and approved in 1998. The revision was reballoted and recirculated, and comments were resolved in 1998. The standard obtained the necessary approval rate during balloting and was submitted to the IEEE-SA Standards Board, which approved it in December 1998.

Purpose

Software quality is the degree to which software possesses a desired combination of attributes. This desired combination of attributes shall be clearly defined; otherwise, assessment of quality is left to intuition. For the purpose of this standard, defining software quality for a system is equivalent to defining a list of software quality attributes required for that system. In order to measure the software quality attributes, an appropriate set of software metrics shall be identified.

The purpose of software metrics is to make assessments throughout the software life cycle as to whether the software quality requirements are being met. The use of software metrics reduces subjectivity in the assessment and control of software quality by providing a quantitative basis for making decisions about software quality. However, the use of software metrics does not eliminate the need for human judgment in software evaluations. The use of software metrics within an organization or project is expected to have a beneficial effect by making software quality more visible.

More specifically, the use of this standard's methodology for measuring quality allows an organization to

- Achieve quality goals;
- Establish quality requirements for a system at its outset;
- Establish acceptance criteria and standards;
- Evaluate the level of quality achieved against the established requirements;
- Detect anomalies or point to potential problems in the system;
- Predict the level of quality that will be achieved in the future;
- Monitor changes in quality when software is modified;
- Assess the ease of change to the system during product evolution;
- Validate a metric set.

To accomplish these aims, both process and product metrics should be represented in the system metrics plan.

The following is a list of major changes from the previous edition:

- a) This revision elaborates on the context in which validation is to be interpreted—validating metrics with respect to a quality factor (e.g., demonstrating a statistical relationship between a complexity metric and defect count for the purpose of predicting defects from complexity) for a given application as opposed to a universal validation of the metrics for all applications. An informative annex (Annex B of this standard), giving sample metric validation calculations, is also included.
- b) Due to the policy of the IEEE Software Engineering Standards Committee (SESC) to provide major informational items in the form of SESC approved books, as opposed to putting this information in standards, the annex that contained the case studies (Annex C of IEEE Std 1061-1992) has been deleted. In addition, due to legal and proprietary restrictions on the release of data when IEEE Std 1061-1992 was written, the working group was unable to obtain metric data from industry for the mission critical example. Therefore, it was necessary to use university metric data. Since the publication of IEEE Std 1061-1992, data have become available from applications, such as the Space Shuttle, which will be used in a future book. The book could be used as a companion document to this standard.
- c) The annex that contained an informational item about metrics descriptions and results (Annex B of IEEE Std 1061-1992) has been deleted because many of the metrics and metrics application results are now obsolete. Examples of metrics will be included in the aforementioned book.
- d) The annex that contained example factors, subfactors, and metrics, and that described the relationships among them (Annex A of IEEE Std 1061-1992), has been deleted. The relationships among these are pictured in Figure 1 and discussed in Clause 3 of this standard.
- e) Obsolete references in the bibliography (Annex D of IEEE Std 1061-1992) have been deleted and a list of references (Clause 2 of this standard) has been added. The purpose of the references is to point the user to additional information about key points in the standard. In accord with SESC policy, a metrics bibliography will be provided in a future SESC approved book.
- f) Due to the importance of the goal question metric (GQM) and the practical software measurement (PSM) frameworks, these frameworks have been included in an informative annex (Annex A of this standard).
- g) Normative and informative material have been better distinguished.

Participants

At the time this standard was completed, the Software Quality Metrics Methodology Working Group had the following membership:

Norman Schneidewind, *Chair*

Alain Abran
Julie Barnard
Dick Chiricosta

Celia Modell, *Editor*

J. Philippe Jacquet
Kathy Liburdy
Tomoo Matsubara

Sandra Swearingen
Leonard L. Tripp
Stephanie White

The following members of the balloting committee voted on this standard:

Syed Ali	Jon D. Hagar	Pavol Navrat
H. Ronald Berlack	John Harauz	Donald J. Ostrom
Michael A. Blackledge	William Hefley	Indradeb P. Pal
Juris Borzovs	James H. Heil	John G. Phippen
James E. Cardow	Mark Heinrich	Peter T. Poon
Enrico A. Carrara	David Heron	Kenneth R. Ptack
Keith Chan	Debra Herrmann	Annette D. Reilly
Antonio M. Cicu	John W. Horch	Dennis Rilling
Rosemary Coleman	John O. Jenkins	Andrew P. Sage
W. W. Geoff Cozens	Frank V. Jorgensen	Helmut Sandmayr
Paul R. Croll	Vladan V. Jovanovic	Stephen R. Schach
Geoffrey Darnton	William S. Junk	Hans Schaefer
Taz Daughtrey	George X. Kambic	Norman Schneidewind
Raymond Day	Diana Kang	David J. Schultz
Bostjan K. Derganc	Chris F. Kemerer	Robert W. Shillato
Perry R. DeWeese	Ron S. Kenett	Lynn J. Simms
Harpal Dhama	Judith S. Kerner	Carl A. Singer
Evelyn S. Dow	Robert J. Kierzyk	Alfred R. Sorkowitz
Sherman Eagles	Thomas M. Kurihara	Luca Spotorno
Leo G. Egan	John B. Lane	Julia Stesney
William Eventoff	J. Dennis Lawrence	Fred J. Strauss
Richard E. Fairley	Randal Leavitt	Sandra Swearingen
John W. Fendrich	Stanley H. Levinson	Toru Takeshita
Jay Forster	William M. Lively	Douglas H. Thiele
Kirby Fortenberry	Dieter Look	Booker Thomas
Eva Freund	John Lord	Patricia Trellue
Karol Fruehauf	Tom Lydon	Leonard L. Tripp
Roger U. Fujii	Stan Magee	Glenn D. Venables
Barry L. Garner	Tomoo Matsubara	Udo Voges
Marilyn Ginsberg-Finner	Patrick D. McCray	Scott A. Whitmire
John Garth Glynn	James Bret Michael	Paul A. Wolfgang
Julio Gonzalez-Sanz	Alan Miller	Paul R. Work
Eric Grosse	Celia H. Modell	Natalie C. Yopconka
L. M. Gunther	Charles S. Mooney	Janusz Zalewski
David A. Gustafson	James W. Moore	Geraldine Zimmerman

When the IEEE-SA Standards Board approved this standard on 8 December 1998, it had the following membership:

Richard J. Holleman, *Chair*

Donald N. Heirman, *Vice Chair*

Judith Gorman, *Secretary*

Satish K. Aggarwal	James H. Gurney	L. Bruce McClung
Clyde R. Camp	Jim D. Isaak	Louis-François Pau
James T. Carlo	Lowell G. Johnson	Ronald C. Petersen
Gary R. Engmann	Robert Kennelly	Gerald H. Peterson
Harold E. Epstein	E. G. "Al" Kiener	John B. Posey
Jay Forster*	Joseph L. Koepfinger*	Gary S. Robinson
Thomas F. Garrity	Stephen R. Lambert	Hans E. Weinrich
Ruben D. Garzon	Jim Logothetis	Donald W. Zipse
	Donald C. Loughry	

*Member Emeritus

Yvette Ho Sang
IEEE Standards Project Editor

Contents

1. Overview	1
1.1 Scope.....	1
1.2 Audience	1
1.3 Conformance.....	2
2. Definitions.....	2
3. Software quality metrics framework (informative).....	3
4. The software quality metrics methodology	5
4.1 Establish software quality requirements	5
4.2 Identify software quality metrics	6
4.3 Implement the software quality metrics.....	8
4.4 Analyze the software metrics results	9
4.5 Validate the software quality metrics	10
Annex A (informative) Additional frameworks	14
Annex B (informative) Sample metrics validation calculations	18
Annex C (informative) Bibliography	20

IEEE Standard for a Software Quality Metrics Methodology

1. Overview

This standard is divided into four clauses. Clause 1 provides the scope of this standard. Clause 2 provides a set of definitions. Clause 3 provides an overview of framework for software quality metrics. Clause 4 provides a methodology for software quality metrics. Also in this standard are three annexes that are included for illustrative and reference purposes only.

1.1 Scope

This standard provides a methodology for establishing quality requirements and identifying, implementing, analyzing, and validating process and product software quality metrics. This methodology applies to all software at all phases of any software life cycle.

This standard does not prescribe specific metrics.

1.2 Audience

This standard is intended for those associated with the acquisition, development, use, support, maintenance, and audit of software. The standard is particularly aimed at those measuring or assessing the quality of software.

This standard can be used by the following:

- An acquisition/project manager to identify, define, and prioritize the quality requirements for a system;
- A system developer to identify specific traits that should be built into the software in order to meet the quality requirements;
- A quality assurance/control/audit organization and a system developer to evaluate whether the quality requirements are being met;
- A system maintainer to assist in implementing modifications during product evolution;
- A user to assist in specifying the quality requirements for a system.

1.3 Conformance

An application of a software quality metrics methodology conforms to this standard if all required provisions, identified by the use of the verb *shall*, are implemented.

2. Definitions

For the purposes of this standard, the following terms and definitions apply. IEEE Std 100-1996 and IEEE Std 610.12-1990 should be referenced for terms not defined in this clause.

2.1 attribute: A measurable physical or abstract property of an entity.

2.2 critical range: Metric values used to classify software into the categories of acceptable, marginal, or unacceptable.

2.3 critical value: Metric value of a validated metric that is used to identify software that has unacceptable quality.

2.4 direct metric: A metric that does not depend upon a measure of any other attribute.

2.5 direct metric value: A numerical target for a quality factor to be met in the final product. For example, mean time to failure (MTTF) is a direct metric of final system reliability.

2.6 measure: (A) A way to ascertain or appraise value by comparing it to a norm. (B) To apply a metric.

2.7 measurement: The act or process of assigning a number or category to an entity to describe an attribute of that entity. A figure, extent, or amount obtained by measuring.

2.8 metric: *See: software quality metric.*

NOTE—The term *metric* is used in place of the term *software quality metric* in this standard.

2.9 metrics framework: A decision aid used for organizing, selecting, communicating, and evaluating the required quality attributes for a software system. A hierarchical breakdown of quality factors, quality subfactors, and metrics for a software system.

2.10 metrics sample: A set of metric values that is drawn from the metrics database and used in metrics validation.

2.11 metric validation: The act or process of ensuring that a metric reliably predicts or assesses a quality factor.

2.12 metric value: A metric output or an element that is from the range of a metric.

2.13 predictive metric: A metric applied during development and used to predict the values of a software quality factor.

2.14 predictive metric value: A numerical target related to a quality factor to be met during system development. This is an intermediate requirement that is an early indicator of final system performance. For example, design or code errors may be early predictors of final system reliability.

2.15 process metric: A metric used to measure characteristics of the methods, techniques, and tools employed in developing, implementing, and maintaining the software system.