



IEEE Standard Classification for Software Anomalies

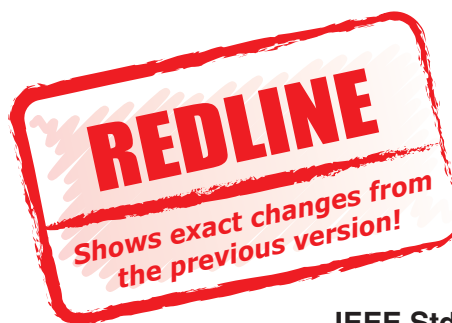
IEEE Computer Society

Sponsored by the
Software & Systems Engineering Standards Committee

1044TM

IEEE
3 Park Avenue
New York, NY 10016-5997, USA

7 January 2010



IEEE Std 1044TM-2009
(Revision of
IEEE Std 1044-1993)

This is a Redline Document produced by Techstreet, a business of Thomson Reuters.

This document is intended to provide users with an indication of changes from one edition to the next. It includes a full-text version of the new document, plus an indication of changes from the previous version.

Redlines are designed to save time and improve efficiencies by using the latest software technology to find and highlight document changes. More professionals are using valuable new technologies like redlines, to help improve outcomes in a fastpaced global business world.

Because it may not be technically possible to capture all changes accurately, it is recommended that users consult previous editions as appropriate. In all cases, only the current base version of this publication is to be considered the official document.

Redline Processing Notes:

1. ~~Red Text~~ - Red strikethrough text denotes deletions.
2. Blue Text - Blue underlined text denotes modifications and additions.

IEEE Standard Classification for Software Anomalies

Sponsor

Software & Systems Engineering Standards Committee

of the

IEEE Computer Society

Approved 9 November 2009

IEEE-SA Standards Board

Abstract: This standard provides a uniform approach to the classification of software anomalies, regardless of when they originate or when they are encountered within the project, product, or system life cycle. Classification data can be used for a variety of purposes, including defect causal analysis, project management, and software process improvement (e.g., to reduce the likelihood of defect insertion and/or to increase the likelihood of early defect detection).

Keywords: anomaly, bug, classification, defect, error, failure, fault, problem

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

Copyright © 2010 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 7 January 2010. 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.

CMMI and Capability Maturity Model Integrated are registered trademarks in the U.S. Patent & Trademark Office, owned by the Carnegie Mellon Software Engineering Institute.

UML is a registered trademark in the U.S. Patent & Trademark Office, owned by the Object Management Group.

PDF: ISBN 978-0-7381-6114-3 STD95995
Print: ISBN 978-0-7381-6115-0 STDPD95995

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. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied “**AS IS**.”

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, or every ten years for stabilization. When a document is more than five years old and has not been reaffirmed, or more than ten years old and has not been stabilized, 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.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon his or her independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

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 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. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

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. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required. Comments and recommendations on standards, and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

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; +1 978 750 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 1044-2009, IEEE Standard Classification for Software Anomalies.

This standard provides a uniform approach to the classification of software anomalies, regardless of when they originate or when they are encountered within the project, product, or system life cycle. Classification data can be used for a variety of purposes, including defect causal analysis, project management, and software process improvement (e.g., to reduce the likelihood of defect insertion and/or to increase the likelihood of early defect detection).

Collecting the data described in this standard provides valuable information that has many useful applications. It is also well documented that the earlier within the software life cycle a problem is discovered, the cheaper, and often easier, it is to fix. This encourages the use of tools, techniques, and methodologies to find problems sooner. Standard anomaly data are necessary to evaluate how well these tools, techniques, and methodologies work. These data can also identify when in a project's life cycle most problems are introduced. Distinctions between enhancements and problems in the software help make the decisions as to which anomalies are addressed first, categories of funding, and so on. Anomaly data can also assist in the evaluation of quality attributes such as reliability and productivity.

Having a standard way of classifying software anomalies is important. First, it enables insight into the types of anomalies that organizations produce during development of their products. This information is a rich source of data that can be used during the execution of a project and for process improvement. Analytical techniques such as Orthogonal Defect Classification and causal analysis depend on classification of anomalies to identify root causes and to help determine means to prevent their recurrence. Process improvement frameworks such as Capability Maturity Model Integrated® (CMMI®)^a promote the need for detailed understanding of process performance and product quality. The classification of anomalies allows the development of profiles of anomalies produced by various development processes as one indicator of process performance.

Second, having a standard way to classify anomalies enables better communication and exchange of information regarding anomalies among developers and organizations. Unfortunately, people frequently associate different meanings with the same words and/or use different words to mean the same thing. Similarly, if software systems are to communicate (i.e., exchange data) effectively regarding anomalies, they must share a common logical (if not physical) data model. Data exchange may still be possible via some mapping or translation method if the same data elements are named differently in one system as compared with another, but each system must at least recognize and implement the same conceptual objects/entities, relationships, and attributes.

This standard is based on several concepts and definitions that must be clearly understood prior to its use. These are discussed briefly in the following paragraphs. Formal definitions can be found in Clause 2, and it is advisable to read them before proceeding.

The word “anomaly” may be used to refer to any abnormality, irregularity, inconsistency, or variance from expectations. It may be used to refer to a condition or an event, to an appearance or a behavior, to a form or a function. The 1993 version of IEEE Std 1044™ characterized the term “anomaly” as a synonym for error, fault, failure, incident, flaw, problem, gripe, glitch, defect, or bug, essentially deemphasizing any distinction among those words. Such usage may be common practice in everyday conversation where the inherent ambiguity is mitigated by the richness of direct person-to-person communication, but it is not conducive to effective communication by other (especially asynchronous) methods. Because a term with such a broad meaning does not lend itself to precise communication, more specific terms are defined and used herein to refer to several more narrowly defined entities. Each entity has associated with it a name, a

^a CMMI and Capability Maturity Model Integrated are registered trademarks in the U.S. Patent & Trademark Office, owned by the Carnegie Mellon Software Engineering Institute. This information is provided for the convenience of users of this standard and does not constitute an endorsement by the IEEE of these products.

definition, a set of attributes, and a set of relationships to other entities. These entities are depicted in Figure 1 and again (using a different notation) in Figure 2, and their definitions can be found in Table 3. The relationships between key entities are modeled in the form of an entity relationship diagram (ERD) in Figure 1 and again in the form of a Unified Modeling Language (UML®)^b class diagram in Figure 2. A detailed explanation of these widely used modeling notations is outside the scope of this standard; however, a brief description of each is provided below the corresponding diagram. Text descriptions of the relationships are available in Table 1, so a complete understanding of the diagrams is not essential to understanding this standard. The entities “Failure,” “Defect,” and “Fault” within the area labeled “IEEE 1044 Scope” in Figure 1 and Figure 2 are the subject of this standard, and the other entities in the diagrams are outside the scope of this standard. Faults are considered a subtype of defect and as such are classified using the same attributes as defects (see Table 4).

To increase flexibility and allow organizations to adapt the classification to their own life cycles and purposes, the following changes have been made to the previous edition of this standard:

- Defining key terms and the relationships between their underlying concepts more precisely
- Not specifying a mandatory set of values for anomaly attributes
- Not specifying a classification process

Several concepts and definitions must be clearly understood before using this standard, so it is highly advisable to review 1.1 and Clause 2 carefully before proceeding to Clause 3. The classification attributes defined in the standard are normative (mandatory), whereas the sample classification attribute values are only informative (optional).

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of

^b UML is a registered trademark in the U.S. Patent & Trademark Office, owned by the Object Management Group.

amendments, corrigenda, or errata, visit the IEEE Standards Association Web site at <http://ieeexplore.ieee.org/xpl/standards.jsp>, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA Web site at <http://standards.ieee.org>.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: <http://standards.ieee.org/reading/ieee/updates/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: <http://standards.ieee.org/reading/ieee/interp/index.html>.

Patents

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 is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this standard was submitted to the IEEE-SA Standards Board for approval, the IEEE 1044 Working Group had the following membership:

David Zubrow, *Chair*
Mark Baldwin, *Vice Chair*

Maryam Asdjodi-Mohadjer
Karen Butler
David Card

Robert Douglas
John Gaffney
Mark Hadley
Jim Kubeck

Celia Modell
Sharon Rohde
Virginia Slavin

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Ed Addario	Andre Fournier	James Moore
Johann Amsenga	David Friscia	Mark Paulk
Butch Anton	Gregg Giesler	Miroslav Pavlovic
Angela Anuszewski	Randall Groves	Ulrich Pohl
Chris Bagge	John Harauz	Annette Reilly
Bakul Banerjee	Mark Henley	Robert Robinson
Zulema Belyeu	Rutger A. Heunks	Terence Rout
H. Stephen Berger	Werner Hoelzl	Randall Safier
Juris Borzovs	Atsushi Ito	Bartien Sayogo
Pieter Botman	Mark Jaeger	Robert Schaaf
Juan Carreon	Cheryl Jones	Hans Schaefer
Keith Chow	Piotr Karocki	David J. Schultz
Daniel Conte	Rameshchandra Ketharaju	Stephen Schwarz
Paul Croll	Dwayne Knirk	Gil Shultz
Geoffrey Darnton	Ronald Kohl	James Sivak
Giuseppe De Francesco	Thomas Kurihara	Thomas Starai
Terry Dietz	George Kyle	Walter Struppler
Thomas Dineen	Susan Land	Gerald Stueve
Antonio Doria	Dewitt Latimer	Marcy Stutzman
Richard Doyle	David J. Leciston	Thomas Tullia
Edward Dudash	Daniel Lindberg	Vincent Tume
Scott Duncan	Vincent Lipsio	John Walz
Sourav Dutta	Carol Long	P. Wolfgang
Carla Ewart	Faramarz Maghsoodlou	Paul Work
Yaacov Fenster	Edward McCall	Oren Yuen
Andrew Fieldsend	William Milam	Janusz Zalewski

When the IEEE-SA Standards Board approved this standard on 9 November 2009, it had the following membership:

Robert M. Grow, *Chair*
Thomas Prevost, *Vice Chair*
Steve M. Mills, *Past Chair*
Judith Gorman, *Secretary*

John Barr	Alexander Gelman	David J. Law
Karen Bartleson	Jim Hughes	Ted Olsen
Victor Berman	Richard H. Hulett	Glenn Parsons
Ted Burse	Young Kyun Kim	Ronald C. Petersen
Richard DeBlasio	Joseph L. Koepfinger*	Narayanan Ramachandran
Andy Drozd	John Kulick	Jon Walter Rosdahl
Mark Epstein		Sam Sciacca

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Howard L. Wolfman, *TAB Representative*
Michael Janezic, *NIST Representative*
Satish K. Aggarwal, *NRC Representative*

Lisa Perry
IEEE Standards Program Manager, Document Development

Malia Zaman
IEEE Standards Program Manager, Technical Program Development

Contents

1. Overview	1
1.1 Scope	1
1.2 Purpose	1
1.3 Field of application.....	1
2. Definitions	5
3. Classification	5
3.1 Classification process	5
3.2 Defect classification	5
3.3 Failure classification.....	6
Annex A (informative) Example values for attributes.....	8
Annex B (informative) Classification examples.....	11
Annex C (informative) Bibliography.....	14

IEEE Standard Classification for Software Anomalies

IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection in all circumstances. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.

1. Overview

1.1 Scope

This standard provides for the core set of attributes for classification of failures and defects. It is recognized that there are other attributes of failures or defects that are of unique value to specific applications or business requirements. This standard is applicable to any software (including operating systems, database management systems, applications, testware, firmware, and embedded software) and to any phase of the project, product, or system life cycle by which the software is developed, operated, and sustained. Classification attributes are unaffected by the choice of life cycle model, and that choice is outside the scope of this standard. Some tailoring of classification attribute values based on the chosen life cycle is expected and consistent with the intent of this standard.

~~This standard is applicable to any software, including critical computer software, commercial applications, system software, support software, testware, and firmware during any phase of a system's life cycle. This standard defines the minimum requirements for classifying anomalies, as well as providing additional classifications for projects requiring greater detail. The mandatory classifications are the minimum requirements necessary to establish a standard terminology for anomalies within or between projects and organizations~~

~~To accomplish the classification task, this standard documents the following subjects:~~

- ~~a) Definitions of terms not provided in IEEE Std 610.12-1990⁺~~
- ~~b) A basic process (sequence of steps) for classifying and establishing categories of anomalies relating to software products and providing related data and information~~
- ~~c) A standard set of categories and classifications, either mandatory or optional~~
- ~~d) A list of supporting data items~~

1.1 Background

This standard is based on several definitions, concepts, and processes that need to be understood prior to its use. These are discussed in the following paragraphs. Formal definitions can be found in clause 3.

A key term in this standard is anomaly. An anomaly is any condition that departs from the expected. This expectation can come from documentation (requirements specifications, design documents, user documents, standards, etc.) or from someone's perceptions or experiences. An anomaly is not necessarily a problem in the software product; it may be manifesting correct behavior in which case changing the software would be an enhancement. An anomaly may also be caused by something other than the software. For reasons of semantics, use of the word *anomaly* is preferred over the words *error*, *fault*, *failure*, *incident*, *flaw*, *problem*, *gripe*, *glitch*, *defect*, or *bug* throughout this standard because it conveys a more neutral connotation.

The methodology of this standard is based on a process (sequence of steps) that pursues a logical progression from the initial recognition of an anomaly to its final disposition. Each step interrelates with and supports the other steps. This process is graphically displayed in figure 1, and a complete discussion of the classification process is in 4.1. This standard does not attempt to enforce any particular anomaly processing procedure other than to identify the basic processes an anomaly goes through. It is expected that users will modify the process based on their organization's procedures. Subclause 4.2 describes the classification scheme and supporting data items useful for identifying, tracking, and resolving anomalies.

1.2 Purpose

The purpose of this standard is to define a common vocabulary with which different people and organizations can communicate effectively about software anomalies and to establish a common set of attributes that support industry techniques for analyzing software defect and failure data.

1.3 Field of application

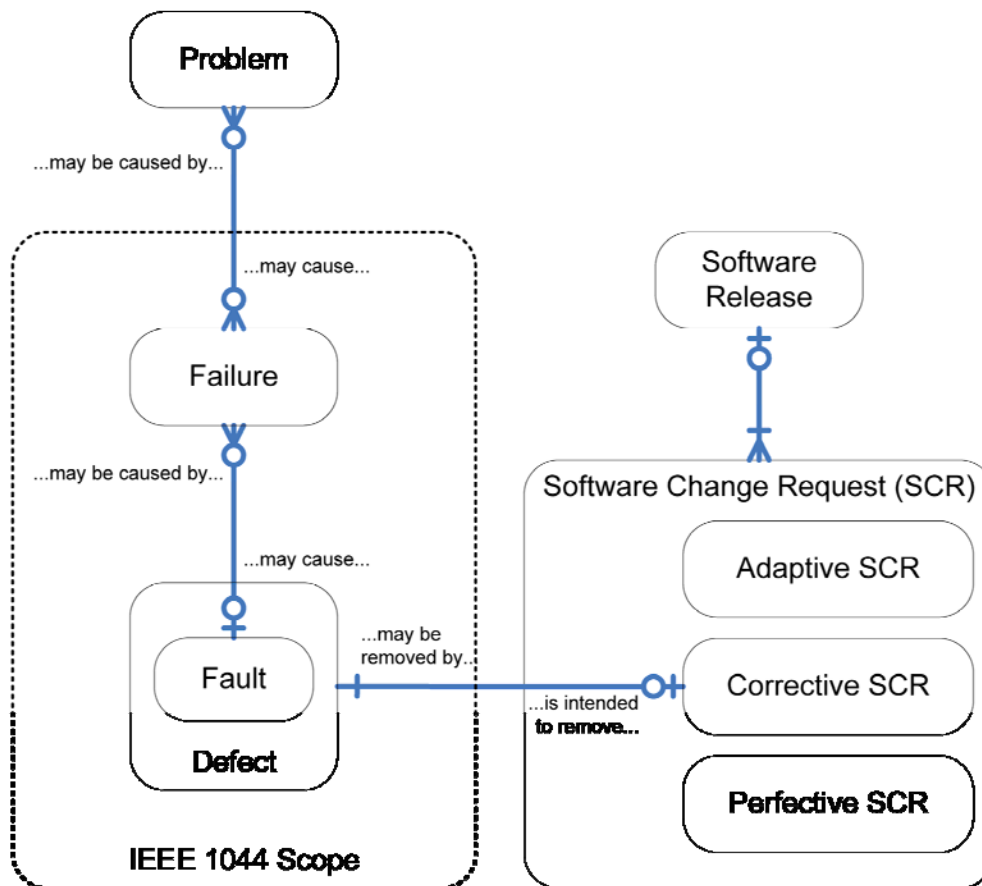
As depicted in Figure 1¹ and Figure 2, problems may be precursors to failure recognition. These are the conditions by which a user recognizes that the software is performing in an undesirable manner. Similarly,

¹ Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

actions taken in response to a failure and fault may be documented as a change request. The classification of problems and change requests is outside of the scope of this standard.

Understanding the scope of this standard depends on understanding the definitions in Clause 2, so it is advisable to read them before proceeding. Scope understanding is also dependent on understanding the relationship among several conceptual entities, which are depicted graphically in Figure 1 and Figure 2 and described textually in Table 1.

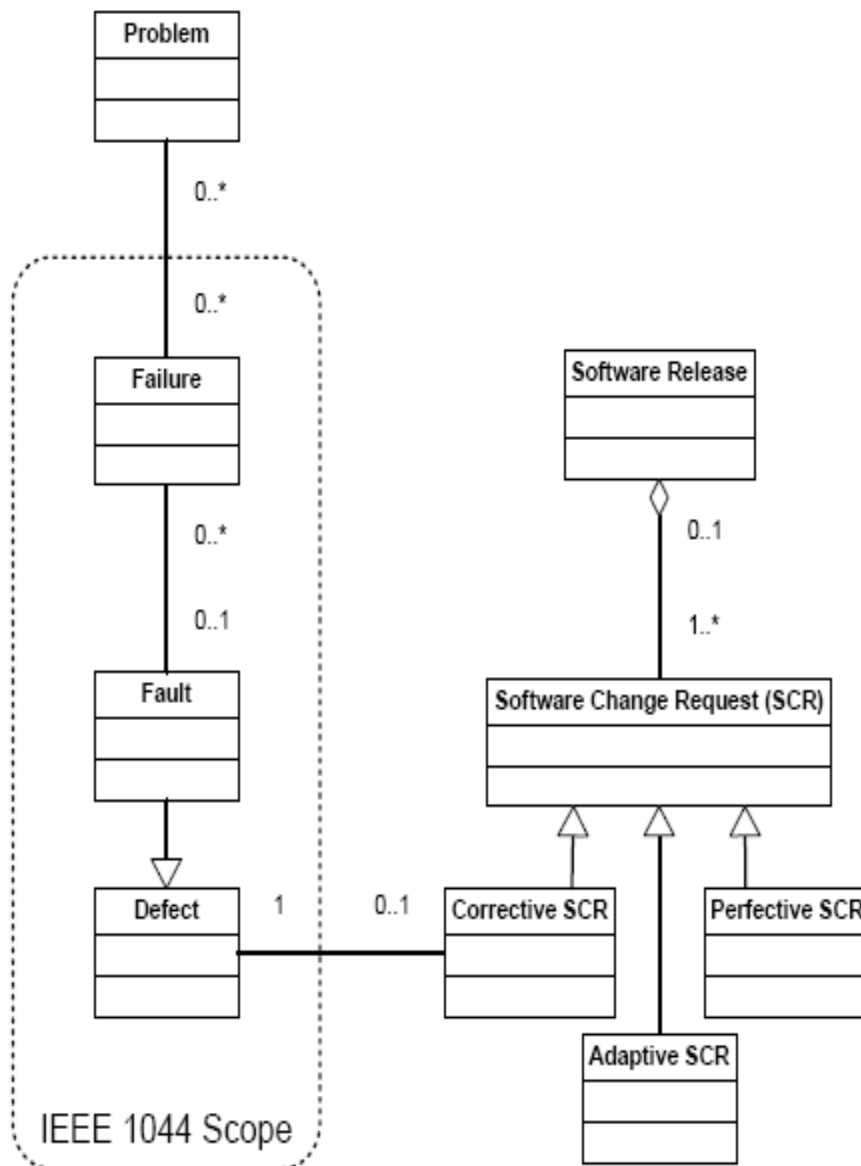
Although software change request (SCR) and software release are not addressed within this standard, they are included in the diagrams to help clarify the scope. As examples, these could have been shown in several different configurations. The one used here is the relatively simple case in which a defect may be associated with a single corrective SCR and each SCR may be associated with, at the most, a single release.



NOTE 1—The rounded rectangles represent entities (things of interest) and the lines connecting the rectangles represent relationships between entities. The symbols at the line ends indicate the number of entities at the end of the line. An open circle near a line end indicates that as few as zero are permitted (i.e., participation is optional); the absence of the circle indicates at least one is required (i.e., participation is mandatory). The three-legged “crows feet” symbol indicates that many entities are permitted to participate; the absence of the crows feet symbol indicates that no more than one entity may participate. A rounded rectangle appearing within another rounded rectangle indicates a parent-child relationship, wherein the contained entity is a subtype of the containing entity (supertype). The relationships represented graphically in this diagram are further described in Table 1.

NOTE 2—This diagram is not intended to mandate a particular notational methodology, and it is not intended as a schema for a database.

Figure 1—Relationships modeled as an entity relationship diagram



NOTE 1—The rectangles represent object classes (things of interest), and the lines connecting the rectangles represent relationships between classes. The three sections within each rectangle contain (from top to bottom) the name, attributes, and methods/operations of the corresponding class. Because the primary focus of this diagram is the relationships, only the class name is included. The methods are outside the scope of this standard, and the attributes are listed and defined in the clauses that follow. The numbers beside the lines indicate the multiplicity of the relationship, with “1” meaning exactly one, “0..1” meaning zero or one, “1..*” meaning one or more, and “0..*” meaning zero, one, or more. Lines with an open triangle on one end indicate a generalization (parent–child) relationship between a supertype class and the subtype class at the other end. Lines with a diamond at the end indicate that more than one change request may be included in one release. The relationships represented graphically in this diagram are further described in Table 1.

NOTE 2—This diagram is not intended to mandate a particular notational methodology, and it is not intended as a schema for a database.

Figure 2 —Relationships modeled as a UML class diagram