IEEE Guide to Classification for Software Anomalies

Sponsor

Software Engineering Committee
of the
IEEE Power Engineering Society

Approved December 12, 1995

IEEE Standards Board

Approved 1 August, 1996

American National Standards Institute

Abstract: This guide provides supporting information to assist users applying the IEEE Std 1044-1993, IEEE Standard Classification for Software Anomalies, to decide whether to conform completely to or just extract ideas from IEEE Std 1044.1. This guide will enable users of IEEE Std 1044-1993 to implement and customize IEEE Std 1044-1993 for their organization in an effective and efficient manner.

Keywords: anomaly, category, classification, software, supporting data item

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

ISBN 1-55937-697-X

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE 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 technical 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 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 all 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.

Introduction

(This introduction is not a part of IEEE Std 1044.1-1995, IEEE Guide to Classification for Software Anomalies.)

On most projects, some formality in reporting anomalies allows you to record the facts and impressions about an anomaly when it is encountered so that it can be understood by someone at a later date. Typically, information about what happened, how it happened, and the environment in which it happened are put into the anomaly reports. IEEE Std 1044-1993 calls this information “supporting data items.” Armed with this information, the anomaly can be reproduced, a fix can be applied, and the new program can be verified not to have the same anomaly symptoms. Many anomaly tracking information systems of this type are well documented, have been used successfully for years, and are thought of as good anomaly tracking systems.

However, when an organization begins to mature, the universal software quality question is asked: “How best can we get rid of the bugs in our products or prevent them from happening?” To effectively answer this question, you need more information about the anomalies that have been found in the software. Knowing the date, submitter, software version, and module an anomaly was reported against usually does not point in the direction of problem areas. The information that will facilitate the discovery of “common errors made,” “most effective activity to find anomalies,” “the place where anomalies were first introduced,” and other telling facts about your development process comes from classification analysis of each anomaly reported. This is where application of the IEEE Std 1044-1993 classification scheme will prove invaluable.

Implementing IEEE Std 1044-1993 can lead to better project management decisions, increased depth of data analysis, and improved software development processes. The consistent historical data allows trend analysis through several releases of the same project, several projects, and/or organizations. This historical consistency of data makes it possible to measure the effects of any process changes you implement from release to release, project to project, or organization to organization. Additionally, using a database and change control procedure reduces the labor of tracking changes, analyzing data, and providing an audit trail. This guide provides the alternatives and methods for getting the most out of IEEE Std 1044-1993.

Many benefits of using IEEE Std 1044-1993 could be obtained by using any classification scheme. However, the big advantage of IEEE Std 1044-1993 is the effort saved by not re-inventing and debating yet another bug taxonomy. Also, wide-spread use of IEEE Std 1044-1993 means that eventually industry-wide information will be available to use for comparative purposes or to benchmark the process(es) used in software development.

At the time this guide was completed, the Classification Guide Working Group had the following membership:

Cynthia Brehmer, *Chair*
Jaya R. Carl, *Co-Chair*

Robert Johnson
Brian Lawrence
Richard Evans
Colin James

The following persons were on the balloting committee:

Syed Ali
Man K. Au
John T. Bell
Mordechai Ben-Menachem
H. Ronald Berlack
Christos Bezirtzoglou
William J. Boll
Fletcher Buckley
David W. Burnett
Michael Caldwell
John P. Chihorek
Antonio M. Cicu
Theo Clarke
Sharon Cobb
Virgil Lee Cooper
Geoff Cozens
Gregory T. Daich
Geoffrey Darnton
Taz Daughtrey
Bostjan K. Derganc
C. Einar Dragstedt
Leo G. Egan
William Eventoff
John W. Fendrich
Julian Forster
Kirby Fortenberry
Eva Freund
Adel N. Ghannam
Hiranmay Ghosh
Julio Gonzalez Sanz
Lawrence M. Gunther
David A. Gustafson
John Harauz
Rob Harker
Myron Hecht
Rick Hefner
Gary E. Hendrzak
Umesh P. Hiriyannaiah
John W. Horch
Charles C. Howell
Fabrizio Imelio
Frank P. Jay
David Johnson
Martha Johnson
Vladan V. Jovanovic
William S. Junk
Chris F. Kemerer
Judy Kerner
Robert J. Kierzyk
Motti Y. Klein
Thomas M. Kurihara
John B. Lane
J. Dennis Lawrence
Stanley H. Levinson
Fang Ching Lim
Mancini Loredana
Austin J. Maher
Robert A. Martin
Tomoo Matsubara
Scott D. Matthews
Mike McAndrew
Patrick McCray
Sue McGrath
Jacques Meekel
Glen A. Meldrum
Jerome W. Mersky
James Bret Michael
Celia H. Modell
Charles S. Mooney
James W. Moore
John Musa
Dennis E. Nickle
Mike Ottewill
Gerald L. Ourada
Lalit Mohan Patnaik
Donald J. Pfeiffer
John G. Phippen
Peter T. Poon
Lawrence S. Przybylski
Dennis Rilling
James R. Roberts
R. Waldo Roth
Terence P. Rout
Andrew P. Sage
Helmut Sandmayr
Stephen R. Schach
Hans Schaefer
Peter E. Schilling
Gregory D. Schumacher
Katsutoshi Shintani
David M. Siefert
Raghu P. Singh
Nancy M. Smith
Melford E. Smyre
Jag Sodhi
Alfred R. Sorkowitz
Julia Stesney
Fred J. Strauss
Toru Takeshita
Booker Thomas
Leonard L. Tripp
Mark-Rene Uchida
Theodore J. Urbanowicz
Glenn D. Venables
David W. Vickers
Udo Voges
Dolores Wallace
William M. Walsh
Camille S. White-Partain
Scott A. Whitmire
Paul A.T. Wolfgang
Paul R. Work
Weider D. Yu
Janusz Zalewski
Geraldine Zimmerman
Peter F. Zoll

When the IEEE Standards Board approved this standard on December 12, 1995, it had the following membership:

E. G. “Al” Kiener, *Chair* **Donald C. Loughry,** *Vice Chair*
Andrew G. Salem, *Secretary*

Gilles A. Baril
Clyde R. Camp
Joseph A. Cannatelli
Stephen L. Diamond
Harold E. Epstein
Donald C. Fleckenstein
Jay Forster*
Donald N. Heirman
Richard J. Holleman
Jim Isaak
Ben C. Johnson
Sonny Kasturi
Lorraine C. Kevra
Ivor N. Knight
Joseph L. Koepfinger*
D. N. “Jim” Logothetis
L. Bruce McClung
Marco W. Migliaro
Mary Lou Padgett
John W. Pope
Arthur K. Reilly
Gary S. Robinson
Ingo Rusch
Chee Kiow Tan
Leonard L. Tripp
Howard L. Wolfman

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
Steve Sharkey
Robert E. Hebner
Chester C. Taylor

Angela M. Girardi
IEEE Standards Project Editor

Contents

ANNEX

IEEE Guide to Classification for Software Anomalies

1. Overview

1.1 Purpose

The purpose of this guide is to provide supporting information to assist users who are applying the IEEE Std 1044-1993[1], IEEE Standard Classification for Software Anomalies, whether to conform completely to IEEE Std 1044-1993 or just to extract ideas from it. The guide will enable users of IEEE Std 1044-1933 to implement and customize IEEE Std 1044-1993 for their organization in an effective and efficient manner.

1.2 Scope

This guide accomplishes the following tasks:

a) Describes how to apply IEEE Std 1044-1993
b) Describes what is and is not compliant customization by
   1) Providing examples of use and customization
   2) Providing example definitions of classification terms and situations where different definitions might be applicable
c) Provides some additional explanation of the “supporting data items” from IEEE Std 1044-1993
d) Provides examples of measures that can be derived from the classification data
e) Provides examples of where those metrics are useful
f) Describes methods for analyzing the classification data

[1] Information about references can be found in clause 2.

1.3 Goals

The goals of this guide are as follows:

a) To facilitate the determination of the level of conformance and amount of detail in IEEE Std 1044-1993 that is appropriate to the organization

b) To define the difference between conforming to IEEE Std 1044-1993 and using it as a reference in the organization’s anomaly classification scheme

1.4 Audiences

This guide serves two audiences. The primary audience is those persons working on a project required to use IEEE Std 1044-1993 and looking for information to help implement IEEE Std 1044-1993 on their project. The second audience is comprised of the project managers or organization managers wishing to start or expand an anomaly tracking system and looking for proven methodologies to support that effort.

2. References

This guide shall be used in conjunction with the following publication:

IEEE Std 1044-1993, IEEE Standard Classification for Software Anomalies.[2]

3. Definitions

The IEEE Std 1044-1993 anomaly classification scheme consists of categories and classifications. The classification scheme is hierarchical in nature. Categories, the top level, represent attributes of an anomaly you might want to classify. Classifications within each category are hierarchical lists of choices to describe the attribute.

3.1 anomaly: Any condition that deviates from the expected based on requirements specifications, design documents, user documents, standards, etc. or from someone’s perceptions or experiences. Anomalies may be found during, but not limited to, the review, test, analysis, compilation, or use of software products or applicable documentation.

3.2 category: An attribute of an anomaly to which a group of classifications belongs. A specifically defined division in a system of classification; class.

3.3 classification: A choice within a category to describe the category (attribute) of an anomaly.

3.4 supporting data item: Data used to describe an anomaly and the environment where it was encountered.

[2] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.

4. Getting started

So, how do you use IEEE Std 1044-1993 most effectively? To start, consider whether you want to review it for ideas on enhancing your existing anomaly classification scheme, or whether you intend to implement IEEE Std 1044-1993 in your organization.

The following process can help you determine what makes the most sense for the project, software development method, life cycle, and organization. These steps assume that you have read IEEE Std 1044-1993 and recognize the terminology used in that standard. Use the following steps and questions as a guide to determine the extent to which you want to use IEEE Std 1044-1993 within your organization:

a) What categories are you currently recording? Refer to clause 5 of this guide for a discussion of what "categories" are, the usefulness of each, and which are mandatory for compliance to IEEE Std 1044-1993.
   1) None — go to next step
   2) Some — list them and map them to IEEE Std 1044-1993 (the terms you currently used may be different)

b) Is conformance to IEEE Std 1044-1993 required?
   1) Yes — select mandatory categories, consider each of the optional categories
   2) No — consider each of the categories

c) List the categories to be considered. For each category, ask the following questions:
   1) Can it be classified easily within your organization?
      i) Yes — select it
      ii) No — continue with (2) below
   3) Could you use the data now or at any time in the future? (Take a quick look at clause 10 to get some ideas for how you might want to use the data)
      i) Yes — select it
      ii) No — continue with (3) below
   3) Would you consider hiring a consultant to collect, analyze, and/or produce this information?
      i) Yes — select it
      ii) No— do not select this category

d) For each category selected, decide which ones are essential for your business today and flag them as mandatory. Naturally, if conformance to IEEE Std 1044-1993 is required, then all of IEEE Std 1044-1993's mandatory categories will be mandatory for your organization.

   When you start using the new anomaly tracking system, it is suggested you minimize the mandatory categories for which you require users to enter data. After the organization becomes comfortable with the anomaly database and realizes a need for complete data, then mandate additional categories as appropriate. If you meet resistance to the data collection, asking people after the fact to "fill in the blanks" encourages them to think about anomaly classification. Knowing how the data will be analyzed and applied also encourages filling in the information—simply knowing that the data will be analyzed (that the effort of providing the information is not futile) can provide sufficient encouragement.

e) For each category selected, list the classifications to be used by your organization. Consider the following, and refer to clause 6 of this guide for a complete discussion of classifications, how to use them, how to define them, how to select them, and how to augment them.

   1) Start with the highest-level classifications

   2) Classify a set of anomalies using the selected categories and the high-level classifications

   3) Does this high-level scheme make sense to you?

4) Add detail (lower hierarchy of classifications) to the categories that are not sufficiently differentiated

5) Create/invent new classifications within the categories for special cases found in your organization and projects

f) Document the categories. Define what the organization means by each category, using the organization's terminology and references. Use clause 5 and annex A of this guide to get started on the definitions.

g) Document the classifications. Write sentences or paragraphs to describe exactly what the classifications mean to your organization, using your organization's terminology. Use 6.1 and annex A of this guide to get started on the definitions.

h) If conformance or reference to IEEE Std 1044-1933 is important to you, document your implementation with respect to IEEE Std 1044-1993. Produce a traceability matrix (similar to that shown in table 12) and provide a conformance rationale.

i) The supporting data items complete the anomaly tracking system. Table 13 represents those pieces of information that may be used to isolate, fix, and track the anomaly. Read through the supporting data items included in IEEE Std 1044-1993 to determine if there are any that sound like good ideas. Remember, there is no requirement in IEEE Std 1044-1993 to use any of the supporting data items. They are listed merely as seeds to get you thinking. Clause 7 of this guide discusses the supporting data items with a slightly different perspective from IEEE Std 1044-1993 in order to help you define which supporting data items to collect.

j) Document all new supporting data items you are adding to your anomaly tracking system. Again, use your organization's terminology and pertinent examples.

k) For each category, identify when during your anomaly tracking process the data is available. Clause 8 of this guide describes examples of anomaly processing, and suggests when during that process (from a variety of perspectives) to collect classification data.

l) Determine how you are going to incorporate the classification of data into your anomaly tracking process. Clause 9 of this guide describes several different methodologies you may be using or may want to consider.

m) Plan how you are going to use the data and how you are going to inform others of analysis performed on the data. This will help you to answer the question in step c.2) above, and you plan for getting the most out of using the classification scheme. Clause 10 of this guide describes valid statistical analysis and a variety of ways in which you can use the classification data to improve your project management, process improvement, and product assessment. Collect the classification data to answer the questions that are most important.

n) Provide training on the classification scheme to your users and to management. Train your users so they are starting with the same understanding. Make sure they know where to find definitions, and make sure they know how you are planning to use the data. This is important to gain their buy-in, to motivate them to take more interest, and to be more regular, accurate and complete in their data entry. Train your management on how you plan to analyze and use the data collected. This will give them a greater interest in the data so that they will encourage their staff to be accurate and complete in their data entry.

5. Categories

The mandatory categories established in IEEE Std 1044-1993 are the minimum set of categories needed to establish a common definition and provide common terminology and concepts to communicate among projects, business environments, and personnel. There is probably some overlap with data you currently collect. Implementation of IEEE Std 1044-1993 for compliance shall include documenting the traceability matrix between the IEEE Std 1044-1993 codes and your classification schema. Think of this traceability matrix as a way to audit conformance to IEEE Std 1044-1993.

Compare your current method with IEEE Std 1044-1993, looking for similarities and adding additional classifications to your chosen categories where necessary. Include all of the categories you might use. If others start using some of the categories and get feedback from the analysis of that data, they will want to use more categories or greater levels of classification detail in order to learn more about what is happening. For that reason, start by requiring the use of only your essential categories. The use of additional categories will follow naturally. Using the codes in IEEE Std 1044-1993 (RR100, IV200, AC100, IM300, DP123, etc.), even if strictly internal to the database, will facilitate comparison of data with others in the industry.

5.1 Category meanings

The following lists show each category, its associated process step, and the question that category should answer. According to IEEE Std 1044-1993, impact identification may occur at any time during the four process steps; however, in practice, the impact categories generally fit best into the Recognition and Investigation anomaly processing steps.

The Recognition categories and their meanings are as follows:

Recognition categories	Definition
Product Status	What is the usability of the product with no changes?
Project Activity	What were you doing when the anomaly occurred?
Project Phase	In which life cycle phase is the product?
Repeatability	Could you make the anomaly happen more than once?
Suspected Cause	What do you think might be the cause?
Symptom	How did the anomaly manifest itself?

The Impact categories that might be classified during the Recognition step are the following:

Recognition Impact categories	Definition
Customer Value	How important a fix is to customers?
Mission/Safety	How bad was the anomaly with respect to project objectives or human well-being?
Severity	How bad was the anomaly in more objective engineering terms?

The Investigation categories and their meanings are as follows:

Investigation categories	Definition
Actual cause	What caused the anomaly to occur?
Source	Where was the origin of the anomaly?
Type	What type of anomaly/enhancement at the code level?

The Impact categories that might be classified during the Investigation step are the following:

Investigation impact categories	Definition
Priority	Rank the importance of resolving the anomaly (subjective)
Project cost	Relative effect on the project budget to fix
Project quality/reliability	Impact to the product quality or reliability to make the fix
Project risk	Risk associated with implementing a fix
Project schedule	Relative effect on the product schedule to fix
Societal	Impact to society of implementing the fix

The Action categories and their meanings are as follows:

Action categories	Definition
Corrective action	What to do to prevent the anomaly from happening again
Resolution	What action to take to resolve the anomaly

The Disposition category and its meaning are as follows:

Disposition category	Definition
Disposition	What actually happened to close the anomaly

You may update the impact classifications during any of the steps of the anomaly process, or you might restrict modification of certain Impact categories to certain steps. The partitioning described in the above lists have been found to be practical for assigning classifications in the Impact categories.

5.2 Standard compliance at the category level

To comply with IEEE Std 1044-1993, an implementation shall include all 11 mandatory categories from IEEE Std 1044-1993. Table 1 lists the mandatory categories according to IEEE Std 1044-1993. Table 2 lists the optional categories.

Table 1—Category use required for compliance with IEEE Std 1044-1993

IEEE Std 1044-1993—Mandatory categories	
Code	**Category**
IV100	Actual cause
DP100	Disposition
RR100	Project activity
IM600	Project cost
RR200	Project phase
IM500	Project schedule
AC100	Resolution
IM100	Severity
IV200	Source
RR500	Symptom
IV300	Type

Table 2— Category use optional for compliance with IEEE Std 1044-1993

IEEE Std 1044-1993—Optional categories	
Code	**Category**
AC200	Corrective action
IM300	Customer value
IM400	Mission/safety
IM200	Priority
RR600	Product status
IM800	Project quality/reliability
IM700	Project risk
RR400	Repeatability
IM900	Societal
RR300	Suspected cause

Exactly which categories you determine to be essential within your organization may vary. For example, table 3 shows the categories used by each of four organizations (A–D) who have experimented with IEEE Std 1044-1993. The final column in table 3 compares these organizations' category choices to the categories labeled *mandatory* in IEEE Std 1044-1993.

Table 3—Mandatory categories used by four organizations compared to IEEE Std 1044-1993

IEEE Std 1044-1993 Category	Organization				Mandatory in IEEE Std 1044-1993
	A	B	C	D	
Actual cause	✓	✓	✓	✓	✓
Corrective action					
Customer value			✓	✓	
Disposition	✓	✓		✓	✓
Mission/safety					
Priority			✓	✓	
Product status					
Project activity	✓	✓	✓	✓	✓
Project cost	✓			✓	✓
Project phase	✓	✓	✓	✓	✓
Project quality/reliability					
Project risk					
Project schedule	✓	✓	✓	✓	✓
Repeatability					
Resolution	✓	✓	✓	✓	✓
Severity	✓	✓	✓	✓	✓
Societal					
Source	✓	✓	✓	✓	✓
Suspected cause					
Symptom	✓	✓	✓	✓	✓
Type	✓		✓	✓	✓

In this example, organizations A and D can claim full compliance with IEEE Std 1044-1993 because only they use all categories defined as *mandatory* in IEEE Std 1044-1993.

6. Classifications

There are several ways to customize the IEEE Std 1044-1993 classifications, as follows:

a) Define the classifications
b) Add classifications
c) Ignore classifications
d) Rename classifications

6.1 Establishing classification definitions

IEEE Std 1044-1993 assumes you will provide definitions for the classifications you use. This way you can create definitions to best fit your organization's products. Example definitions are included in annex A to either use as is or to help you get started developing definitions customized to your organization.

The Customer Value field provides a useful illustration of how different definitions apply to different organizations. Table 4 shows the classifications for Customer Value from IEEE Std 1044-1993.

Table 4—Customer Value classifications from IEEE Std 1044-1993

IEEE Std 1044-1993—Customer Value	
Code	**Classification**
IM310	Priceless
IM320	High
IM330	Medium
IM340	Low
IM350	None
IM360	Detrimental

Tables 5 and 6 show two different examples of classification definitions for the same category (Customer Value). An organization selling commercial, off-the-shelf software to many customers might use the definitions for Customer Value shown in table 5, whereas an organization satisfying a single customer might instead use the definitions shown in table 6. Note that in table 6 the number of classifications was reduced from 6 to 4, so technically this example does not comply with IEEE Std 1044-1993 because it does not use all the first level classifications for Customer Value.

Table 5— Example Customer Value classifications for commercial model

Commercial products—Customer Value	
Classification	**Description**
Priceless	The customer cannot continue using the software, or the customer will not buy the software until this anomaly is addressed. (CUSTOMER VALUE = 1)
High	The customer can still use the software, but cannot perform a critical task, or the customer is seriously considering not purchasing the product until this anomaly is addressed. (CUSTOMER VALUE = 2)
Medium	The customer can use the software (a workaround exists) but the defect is very annoying or the customer will purchase on the assurance that the anomaly will be addressed in a future release. (CUSTOMER VALUE = 3)
Low	The customer is mildly interested in the anomaly though the work around is an acceptable alternative, or the customer will purchase without anomaly resolution. (CUSTOMER VALUE = 4)
None	Several customers have indicated conflicting opinions about the anomaly. (CUSTOMER VALUE = 5)
Detrimental	Addressing the anomaly would have a negative effect on existing customers or could limit sales. (CUSTOMER VALUE = 6)

Table 6—Example Customer Value classifications for contract developers

Custom software development—Customer Value	
Classification	**Description**
Critical	An immediate loss of revenue results from a non-conformance to a contract requirement. (CUSTOMER VALUE = 1)
High	A failure to comply with contract requirements or the requirements definition. (CUSTOMER VALUE = 2)
Medium	A failure to comply with organization or project defined specification, procedure or standard. (CUSTOMER VALUE = 3)
Low	All other anomalies or action items. (CUSTOMER VALUE = 4)

6.2 Choosing classifications within a category

Project Phase provides a good example of the detail levels and how the level of detail in the classifications you choose are customizable. Table 7 provides the available classifications for Project Phase according to IEEE Std 1044-1993. The Project Phase classifications are a good example of the hierarchical nature of the classifications. The top levels of the hierarchy are Requirements, Design, Implementation, Test, Operation & Maintenance, and Retirement. Within these classifications, more detailed classifications are listed.

Table 7—Project Phase classifications from IEEE Std 1044 -1993

IEEE Std 1044-1993—Project Phase	
Code	**Classification**
RR210	Requirements
RR211	Concept Evaluation
RR212	System Requirements
RR213	Software Requirements
RR214	Prototype Requirements
RR220	Design
RR221	System Design
RR222	Preliminary Design
RR223	Detail Design
RR224	Prototype Design
RR230	Implementation
RR231	Code
RR232	Unit Test
RR233	Integrate
RR234	Prototype
RR240	Test
RR241	Integration Test
RR242	System Test
RR243	Beta Test
RR244	Prototype Test
RR245	Acceptance Test
RR246	Installation and Checkout
RR250	Operation and Maintenance
RR260	Retirement

However, a commercial organization or smaller project might choose to use the Project Phase classifications shown here in table 8.

Table 8—Example Project Phase classifications for commercial model

Commercial model—Project Phase	
Code	**Classification**
RR210	Requirements (definition)
RR220	Design
RR230	Implementation
RR240	Test
RR250	Maintenance
RR260	Retirement

A Department of Defense project started under [B1] may have chosen to use the Project Phase classifications shown here in table 9.

Table 9—Example Project Phase classifications for DoD-STD- 2167A

DoD 2167A Model—Project Phase	
Code	**Classification**
RR212	System Requirements Analysis/ Design
RR213	Software Requirements Analysis
RR222	Preliminary Design
RR223	Detailed Design
RR231	Coding and CSU Testing
RR233	CSC Integration and Testing
RR241	CSCI Testing
RR242	System Integration and Testing

Another DoD project might use the high level acquisition phases (project life cycle phases) in accordance with [B8] and [B2], as shown in table 10. This is an example of how you may add classifications to a category.

Table 10—Example Project Phase classifications for DoD model

DoD Model—Project Phase	
Code	**Classification**
RR201*	Determination of Mission Need
RR211	Concept Exploration & Definition, Phase 0
RR202*	Demonstration & Validation, Phase 1
RR203*	Source Selection
RR220	Engineering & Manufacturing Development, Phase II
RR230	Production & Deployment, Phase III
RR250	Operations & Support, Phase IV

*Note that codes RR201, RR202, and RR203 are not in IEEE Std 1044-1995. They were created to complete the phases for this lifecycle model example.

A final example is an organization following the Spiral Life Cycle Model. This organization might use the classifications for Project Phase shown here in table 11.

Table 11—Example Project Phase classifications for spiral model

Spiral model—Project Phase	
Code	**Classification**
RR211	Concept Evaluation
RR212	System Requirements
RR214.0-9	Prototype Requirements
RR224.0-9	Prototype Design
RR234.0-9	Prototype Implementation
RR244.0-9	Prototype Test
RR213	Software Requirements
RR221	System Design
RR223	Detail Design
RR231	Code
RR232	Unit Test
RR233	Integrate
RR241	Integration Test

Table 11—Example Project Phase classifications for spiral model *(Continued*)

Spiral model—Project Phase	
Code	**Classification**
RR242	System Test
RR243	Beta Test
RR246	Installation and Checkout
RR250	Operation and Maintenance
RR260	Retirement

Note that each time through the prototyping cycle of the spiral life cycle, the decimal portion of the classification code is incremented.

As you can see, you may want to add some additional classifications to those provided in IEEE Std 1044-1993. However, the classifications provided cover the majority of anomalies encountered during pilot usage of IEEE Std 1044-1993 during its development in a variety of commercial and military organizations.

6.3 Standard compliance at the classification level

For each category implemented from IEEE Std 1044-1993, all the first level classifications for the category shall be included (as well as any other applicable classifications you determine). Some implementations of IEEE Std 1044-1993 will require the addition of classifications not included in IEEE Std 1044-1993. Any additions or extensions to IEEE Std 1044-1993 classifications are within compliance. The additions shall be noted as such when you prepare the compliance document that includes your traceability matrix.

Additionally, the compliance document shall contain a list of all categories and classifications that you have implemented. IEEE Std 1044-1993 assigns a code to each category and classification. Map the codes from IEEE Std 1044-1993 to the items in your implementation. By doing this, you can use your own words to represent a classification while maintaining traceability and compliance to IEEE Std 1044-1993. An example traceability matrix including this information is shown in table 12.

Even if you choose not to implement IEEE Std 1044-1993 fully, a compliance document that describes how IEEE Std 1044-1993 maps to your implementation is strongly recommended (especially where you have changed the names of any of the categories or classifications). This compliance document should state which categories are implemented. If one of the mandatory categories is not used, include a justification. Most importantly, include your traceability matrix, as this is what allows you to compare your classification data with other organizations.

IEEE Std 1044-1993 lists classifications within each category in an order that made sense when IEEE Std 1044-1993 was first drafted and the codes were assigned. As the standard was developed and reviewed, the classification names were modified, new classifications were added, and others were removed from the lists. As a result, the list of classifications for some categories could be reordered to improve the usability of your implementation. See table 12 for an example of how the Project Phase, Symptom, and Priority category classifications can be reordered and renamed to improve the logic of the list.

For example, the Project Phase "RR250 Operation and Maintenance" is often known by a different term, such as "support," "sustaining," "release," or "product." In the IEEE Std 1044-1993 compliance implementation traceability matrix example below (table 12), the Class ID is a unique integer identifier used internally by the database (not seen by the users) while Implementation_name is the text the users see when using the anomaly tracking database.

Table 12— Example IEEE Std 1044-1993 compliance traceability matrix

Category	IEEE Code	Classification	Local Class ID	Implementation_name
Project Phase	RR200	Project Phase	11200	Phase
	RR210	Requirements	11210	Reqs
	n/a	n/a	11270	Dormant
	RR220	Design	11220	Designing
	RR230	Implementation	11230	Coding
	RR240	Test	11240	Testing
	RR250	Operation and Maintenance	11250	Sustaining
	RR260	Retirement	11260	End of life
	n/ a	n/a	11299	Unknown
Symptom	RR500	Symptom	11500	Symptom
	RR570	Perceived Total Product Failure	11570	Total product failure
	RR560	Failed Required Performance	11560	Requirement failed
	RR510	Operating System Crash	11510	OS Crash
	RR580	System Error Message	11580	OS error message
	RR530	Program Crash	11530	Application crashes
	RR520	Program Hung-up	11520	Application hangs
	RR541	Correct input Not Accepted	11541	Good input anomaly
	RR542	Wrong Input Accepted	11542	Bad input anomaly
	RR543	Description Incorrect or Missing	11543	Help or prompt anomaly
	RR544	Parameters Incomplete or Missing	11544	Parameter anomaly
	RR540	Input Problem	11540	Other input anomaly
	RR551	Wrong Format	11551	Output format anomaly
	RR552	Incorrect Results/Data	11552	Output data wrong
	RR554	Spelling/Grammar	11554	Spelling or grammar
	RR553	Incomplete/Missing	11553	Output data missing
	RR555	Cosmetic	11555	Visual nit
	n/a	n/a	11559	Look/feel anomaly

Table 12— Example IEEE Std 1044-1993 compliance traceability matrix *(Continued)*

Category	IEEE Code	Classification	Local Class ID	Implementation_name
	RR550	Output Problem	11550	Other output anomaly
	RR590	Other	11590	Other
	n/a	n/a	11599	Unknown
Priority	IM200	Priority	55200	Implementation priority
	IM210	Urgent	55210	Do it now
	IM220	High	55220	High
	IM230	Medium	55230	Medium
	IM240	Low	55240	Low
	IM250	None	55250	No action
	n/a	n/a	55299	Unknown

In this example, "Unknown" functions as a place-holder for a field not yet classified. This may be more useful than simply leaving the field blank, as some fields may not apply in some cases (and would therefore be left blank), and helps to identify classification work that still needs to be done for a given anomaly.

7. Supporting data items

When an anomaly is recorded many pieces of information are collected. IEEE Std 1044-1993 separates the information into two parts: categories and supporting data items. The category taxonomy is for the information that is both commonly recorded and significant to cross-project analysis. The supporting data items are all the other information gathered during the anomaly's lifetime. Supporting data items are such things as the description of an anomaly, information about the environment where it was encountered, and details about its resolution.

The diagrams of supporting data items provided in IEEE Std 1044-1993 may be confusing. These lists of supporting data items are provided merely to trigger ideas of items you might want to document. There are no relationships between the supporting data items, as implied by the diagrams in IEEE Std 1044-1993. In fact, it may be more accurate to think of the supporting data items as falling into the 5 generalized groups shown in table 13.

As the definition states, supporting data items "describe an anomaly." A plain text description of an anomaly and the circumstances under which it was discovered is the core of every anomaly report. Most anomaly databases include a full text description as well as a single line synopsis frequently used in reports. Consider including a public summary description in your database if anomaly reports are going to be made available to external customers.

Table 13— Groups of supporting data items

Group	Types of data
Text	Description, public summary, one-liner, notes, workaround, proposed fix, documentation modification directives, keywords, running commentary, etc.
Identifiers	Software version, anomaly ID, software serial number, CPU, customer name/number, hardware, network configuration, peripherals, customer's anomaly tracking ID, contract change class, etc.
Measurements	Date encountered, date reported, date resolved, CPU time to failure, hours to fix, hours to test, hours to document, LOC affected, number of modules/files modified, total inspection time, etc.
Pointers	Related anomaly IDs, customer information, documentation affected, submitter login name, symptom log file, test data file, test case, core dump, screen capture, etc.
Administrative	Submitter, responsible engineer, test engineer, state/ phase in anomaly processing, peer reviewer, fix date/build deadline, documentation update deadline, verification required (unit test, inspection, none), etc.

8. Anomaly processing

This clause includes some examples of real-life anomaly processing situations and how they relate to the generic anomaly processing model included in IEEE Std 1044-1993. Figure 1 is a variation on the figure 1 from IEEE Std 1044-1993, representing the same concept in a slightly different way. This figure illustrates the basic steps of the anomaly process as described in IEEE Std 1044-1993. The major process steps are: Recognition, Investigation, Action, and Disposition. Included in these steps are three administrative activities: (a) recording information, (b) classifying attributes of the anomaly, and (c) identifying the impact of various aspects of the anomaly. These administrative activities occur during each of the four major process steps, where classification and impact identification may occur or be changed throughout the process.

8.1 Generic anomaly process

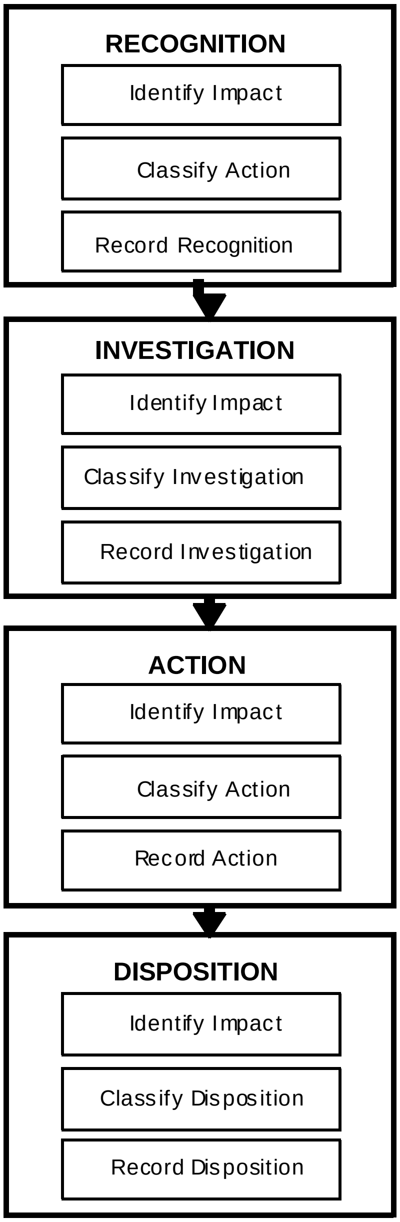


Figure 1—Classification process flow chart-variation from IEEE Std 1044-1993

The classification scheme is divided into categories that apply to each of the Recognition, Investigation, Action, and Disposition process steps.

IEEE Std 1044-1993 recognizes that these four basic steps (Recognition, Investigation, Action, and Disposition) are involved in any change control process, and that a change control process including these four steps is required for effective use of the classification scheme in IEEE Std 1044-1993. IEEE Std 1044-1993 makes no attempt to dictate a change control process; it simply acknowledges that first you must recognize a need for a change, you figure out what to do about it, you do it, then attempt to prevent the same or similar anomalies from happening again, and finally, you give the anomaly a final resting place.

IEEE Std 1044-1993 categories are entered into the anomaly database (or anomaly tracking system) at different phases of the anomaly disposition process. The people responsible for each step in the change control process should enter and update the categories classified during that step. Table 14 shows the anomaly process step where each category is first recorded and later steps where the information may be modified as more data becomes available.

Table 14— Anomaly process step recommendations for recording each category

Category	Recognition	Investigation	Action	Disposition
Actual Cause		Enter	Update	
Corrective Action			Enter	Update
Customer Value	Enter			
Disposition				Enter
Mission/Safety	Enter	Update	Update	Update
Priority	Enter	Update	Update	Update
Product Status	Enter	Update		
Project Activity	Enter			
Project Cost		Enter	Update	Update
Project Phase	Enter			
Project Quality/Reliability		Enter	Update	
Project Risk		Enter	Update	Update
Project Schedule		Enter	Update	Update
Repeatability	Enter	Update		
Resolution			Enter	Update
Severity	Enter	Update		
Societal		Enter		Update
Source		Enter	Update	
Suspected Cause	Enter	Update		
Symptom	Enter	Update		
Type		Enter	Update	

8.2 Specific example of a real anomaly process mapped to the generic

The purpose of the following set of change control process diagrams is to show how a real change control process can be mapped to the basic steps described in IEEE Std 1044-1993 (and reiterated here in figure 1).

Figure 2 shows a complete change control and anomaly processing flow chart. Understanding each step and decision in this diagram is not important, though there is likely to be some similarity with processes you use. What is important is that when this flow chart is converted to a state transition diagram, figure 3, the states through which an anomaly passes can be easily mapped to the four basic steps discussed in IEEE Std 1044-

1993 as shown. When this state transition is further refined to the states through which the majority of anomalies pass, figure 4 is the result. Again, the mapping between this state transition diagram and the process in IEEE Std 1044-1993 is easy to see.

The figures here use the following abbreviations: CMCB is Configuration Management Control Board, dev. is DEVelopment, DOC is DOCumentation, int. is INTegration, INV is INVestigation or study, PRB is Product Review Board, SUBMIT means fix is submitted for integration into source tree, QA is Quality Assurance for analysis and test, and S/W is SoftWare.

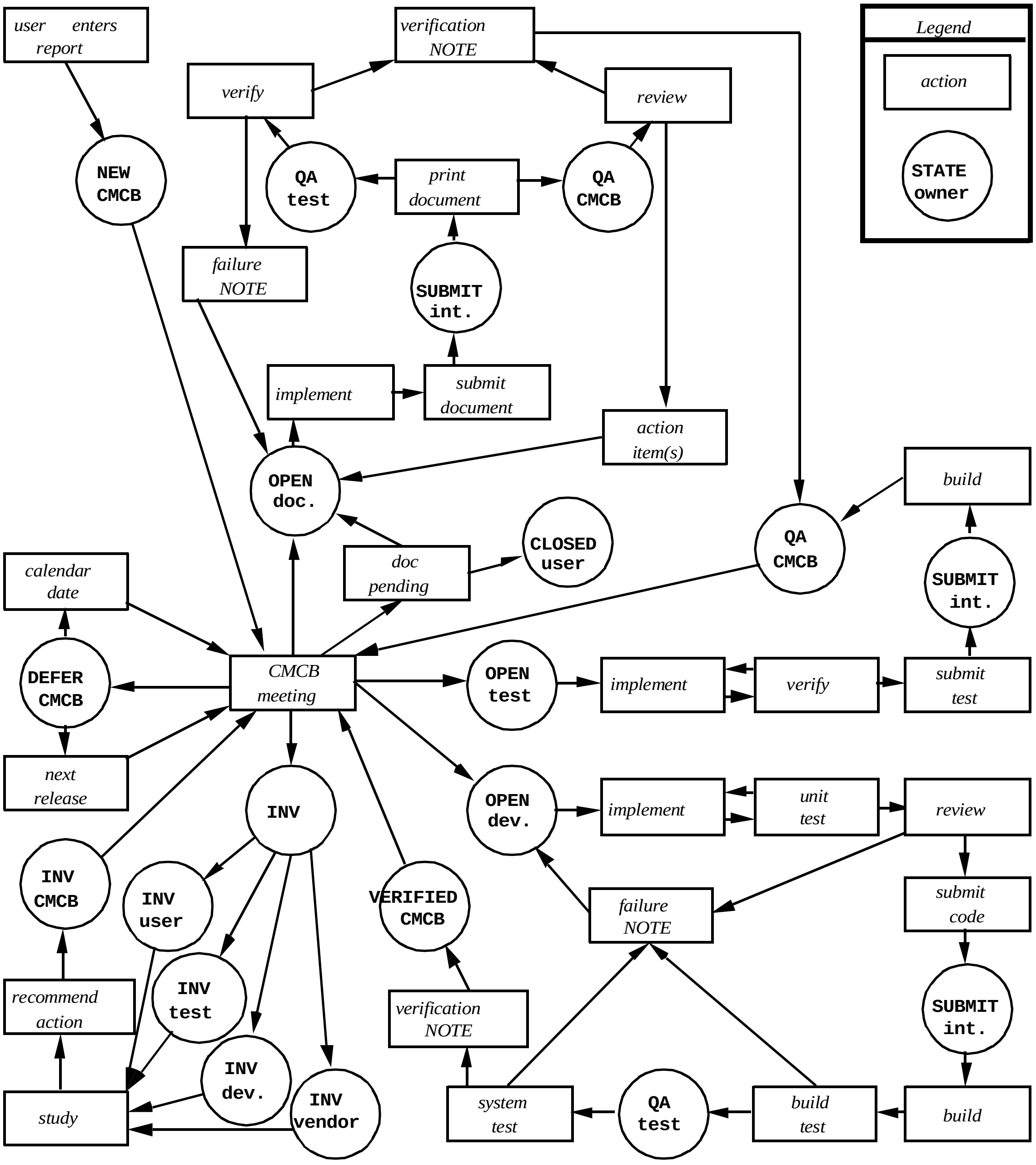


Figure 2—Example of complete change control process flow chart

Figure 3 shows the state diagram resulting from the process flow chart of figure 2, and how the states are mapped to the process steps of figure 1.

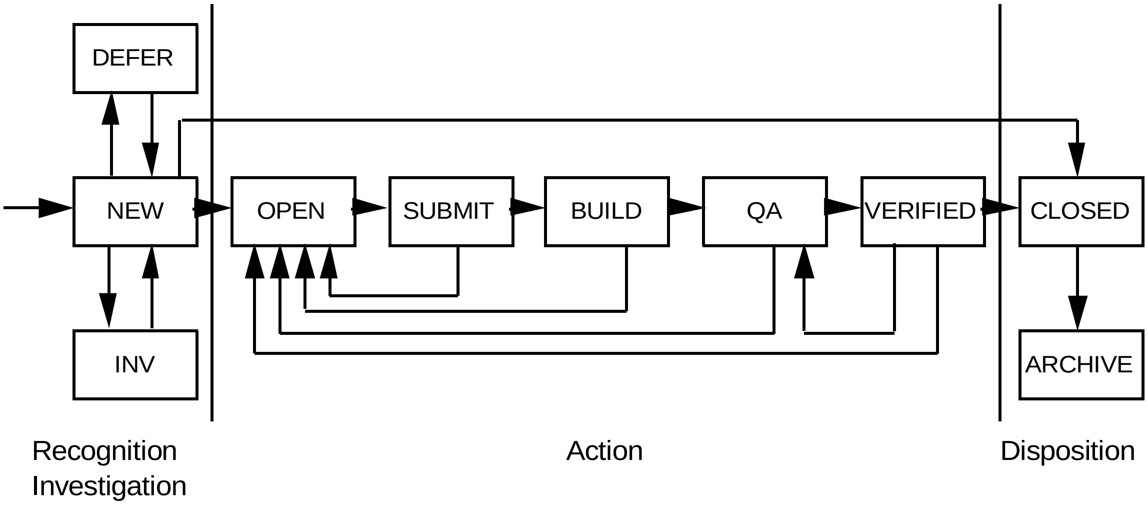


Figure 3—Example of complete STATE transitions

Figure 4 shows the most commonly used state transitions from figure 3, again showing the relationship to IEEE Std 1044-1993.

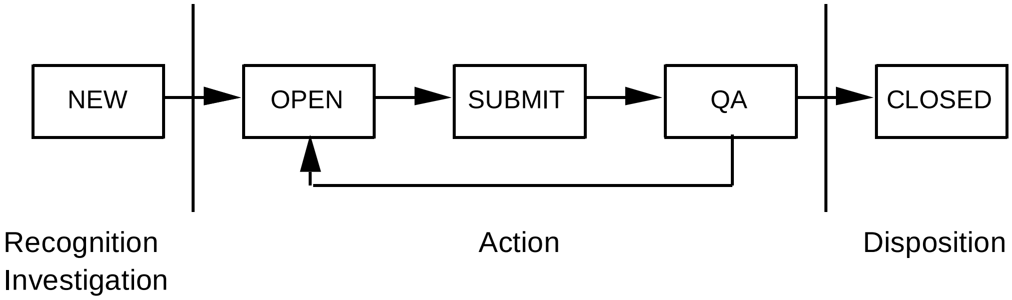


Figure 4—Example of common STATE transitions

9. Methodologies

The ideal time to change classification scheme paradigms is with a new system. If the new system is replacing an existing system, then a well-planned and introduced change to IEEE Std 1044-1993 is relatively painless. If you have no existing system of classification, then introducing IEEE Std 1044-1993 is even easier (from the perspective that you do not have to retrain from an existing classification scheme). IEEE Std 1044-1993 is likely to map to your needs since it is both a shrink-to-fit and, if necessary, expandable model.

9.1 Using a commercial anomaly tracking product

There are many commercial anomaly tracking database products available. They come in many shapes and sizes. There are simple applications that run on PCs, all the way up to client/server systems for multiple users. Most of these can be customized to include the classification scheme, though some are more easily customized than others. The advantages to using a commercial anomaly tracking product are as follows:

a) It is already built.

b) The classification scheme can usually be added.

c) You do not have to use internal resources to build and test an internal tool.

d) The vendor's system may well have a more robust design than what you might identify. The vendor's primary task is to design the tracking system, where your primary focus is on your own products. (The vendor is likely to have more expertise because they have a wider exposure to tracking systems than you do.)

e) The vendor supplies better technical support.

f) The vendor's user groups can provide valuable information on the system you purchased, tracking models, technology, etc.

Disadvantages to using a commercial product are the following:

a) You have to use the product's method of processing anomalies.
b) Adding the classification scheme will probably be a consulting project (adding to the cost). In addition, replacing the classification scheme is often a major change to an existing change control system. To date, no commercial products have an implementation of IEEE Std 1044-1993.
c) It is unlikely that you can implement an unmodified system. High levels of resistance are likely if the model does not match closely enough with existing processing paradigms.
d) Local customizations may introduce version control problems with the vendor's releases.
e) Even though it's already built, there is considerable work to get it running.
f) The commercial product may not be cheaper than building your own with a 4th Generation Language.

9.2 Using an internally-developed anomaly tracking product

You can personally develop your own anomaly tracking database. The advantages to this are the following:

a) You can use the specific anomaly processing method you want.

b) Adding the classification scheme is just a few more fields with specific option lists.

c) You have more control over where to place the classification fields in your interface, such that you encourage accurate data entry.

The disadvantages to building your own are as follows:

a) You have to use the resources to design, build, and test. This may include acquiring entirely different hardware and supporting software than you already have, or falling to the temptation of using existing systems which are not and never will be adequate to support your organization's needs.
b) Development of internal tools is rarely treated with the same level of planning, schedule, resource allocation, and quality as shipping products. While the quality and robustness of anomaly tracking systems can have a profound effect on the productivity of your organization, it is hard to see that when you are just getting ready to plan your system. The tendency is to vastly underestimate the cost of developing and maintaining the system.
c) You may not have the expertise in-house to develop such a system.
d) You must provide your own support.
e) Development may well be just as costly as buying a solution, and may take much longer

9.3 Linking with other tools, processes, and planning documents

There is a growing movement to associate anomaly information with more than source code. Some vendors of configuration management systems and/or call tracking systems include anomaly tracking as a standard feature of their products. This provides relatively seamless relations between anomaly records and artifacts used in building software products such as requirements, designs, plans, and other specifications (such as source code). Also, some anomaly tracking system vendors are implementing links to configuration management systems. Access to data which maps anomaly records to designs, specifications, source code, etc., has tremendous potential benefit for process improvement.

9.4 Using paper versus a database for tracking anomalies

There's no special reason for using a computerized database to track anomalies unless you have a large number of people needing simultaneous access. Many organizations use sheets of paper in a binder. This works well up to the point when you need to share the data outside a group that can effectively get access to your binder. The most compelling reason (other than access) for switching to a database implementation comes when you want to analyze your anomaly data.

Though you can keep your anomaly data and classifications on paper forms (or electronic forms), it is highly recommended that you use an electronic database. Add the categories to whatever database forms you currently use. There are many organizations using the electronic equivalent of paper (e.g., a directory for each project/a subdirectory for each module or program/a file for each anomaly.) All of these single anomaly files use the same template. Programs are then written to extract information for the files and make reports. This is equivalent to having a large filing cabinet with a drawer for each project, a folder for each module and a sheet of paper for each anomaly. The information is systematically recorded and filed, and it can be retrieved, but the storage system does not lend itself to data collection and analysis. A good database design facilitates the collection, abstraction, and analysis of the information it holds. For some projects, a spreadsheet can suffice for the basic database functions and data analysis desired.

The number of your product configurations and the size of your user community are factors to consider when designing your database. For a small number of product configurations, you can handle the tracking of anomalies and their fixes within the same database object. However, with a large or diverse set of product configurations, it becomes much more advantageous to separate the anomaly reports from the fix management. When the anomaly reports are separated from the fix management, you will need to provide bi-directional links between the anomalies and the fix(es) that address them.

10. Data analysis

Analysis of the data you provide through use of IEEE Std 1044-1993 is where the value of following IEEE Std 1044-1993 becomes apparent. The data analysis can be used for project management, process improvement and product assessment. There are three ways to view the data analysis. One way is done on a case-by-case basis where the results of the analysis are immediate. This is the kind of analysis used by project managers for quick feedback on the product development process. Each individual anomaly is classified and then compared to the project plan or development model for immediate insight into the condition of the product. The second way is to look at the data from the entire project perspective to identify possible problem areas within that project. The third way to use the data involves looking at the anomalies across several projects within an organization or even across several organizations to identify organizational problem areas or industry-wide problem areas.

10.1 Statistical analysis

This subclause is intended to provide an overview of reasonable statistical analyses to be used on classification data. It is assumed that persons analyzing classification data will have some statistical analysis experience, tools, and/or other resources to provide additional detail where needed.

Use the level of detail of the classifications that appears the most appropriate based on the number of anomalies you observe on average and your internal processes. Keep in mind that in order to do meaningful data analysis you need to have a statistically significant number of anomalies (~30) assigned to each classification. Therefore, choose the level of detail carefully; too much detail will slow down gathering a statistically significant number of data points. Too little detail can make the data you gather too general to be insightful.

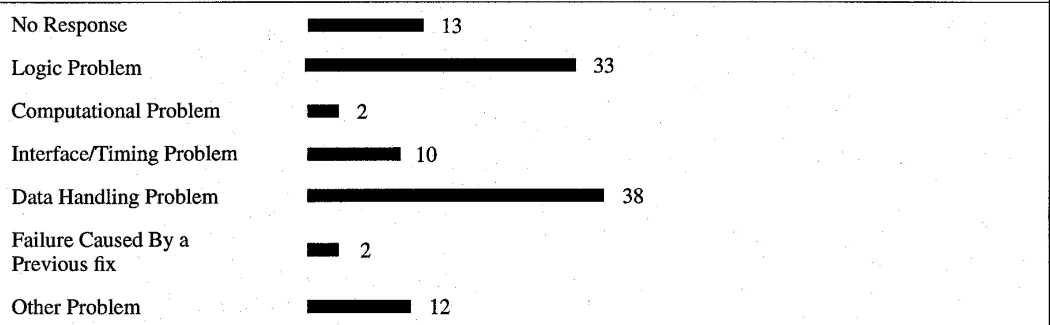
To increase the ease of the statistical analysis, you'll probably want to use a software statistical package. There are several on the market that allow a more complete analysis (including frequencies, Chi-Square, correlation, and others). Averages, standard hypothesis testing (t and z tests), and other parametric statistics are meaningless with classification data, since these data do not follow a rational scale. This precludes the use of averages when you analyze classification data.

The frequency of occurrence in each classification can identify broad problem areas. Although a frequency distribution can be examined to determine which classifications seem most used, a Chi-Square test over all classifications within a category will indicate if the classifications are equally likely, at a given level of significance. The Chi-Square analysis in the following examples assumes that the frequency of each classification is uniform, which is the usual hypothesis of interest. The following examples also use the most commonly used significance level of 0.05 or less to indicate if the classifications frequency distribution is uniform. This statistic may be used to indicate whether or not there is a statistically significant difference between the counts in some of the classifications.

Figure 5 is an example of a frequency analysis with Chi-Square of the category type that you might get from a statistical analysis package. This example does not use the entire list of type classifications included in IEEE Std 1044-1993:

Types of software anomalies observed

Classification	Frequency	Percent	Expected	Residual
No Response	13	11.8	15.71	–2.71
Logic Problem	33	30.0	15.71	17.29
Computational Problem	2	1.8	15.71	–13.71
Interface/timing Problem	10	9.1	15.71	–5.71
Data handling Problem	38	34.5	15.71	22.29
Failure Caused by a Previous fix	2	1.8	15.71	–13.71
Other Problem	12	10.9	15.71	–3.71
Total	110	100.0		



Chi-Square = 77.982 Degrees of Freedom = 6 Significance = 0.000

Figure 5—Frequency analysis example

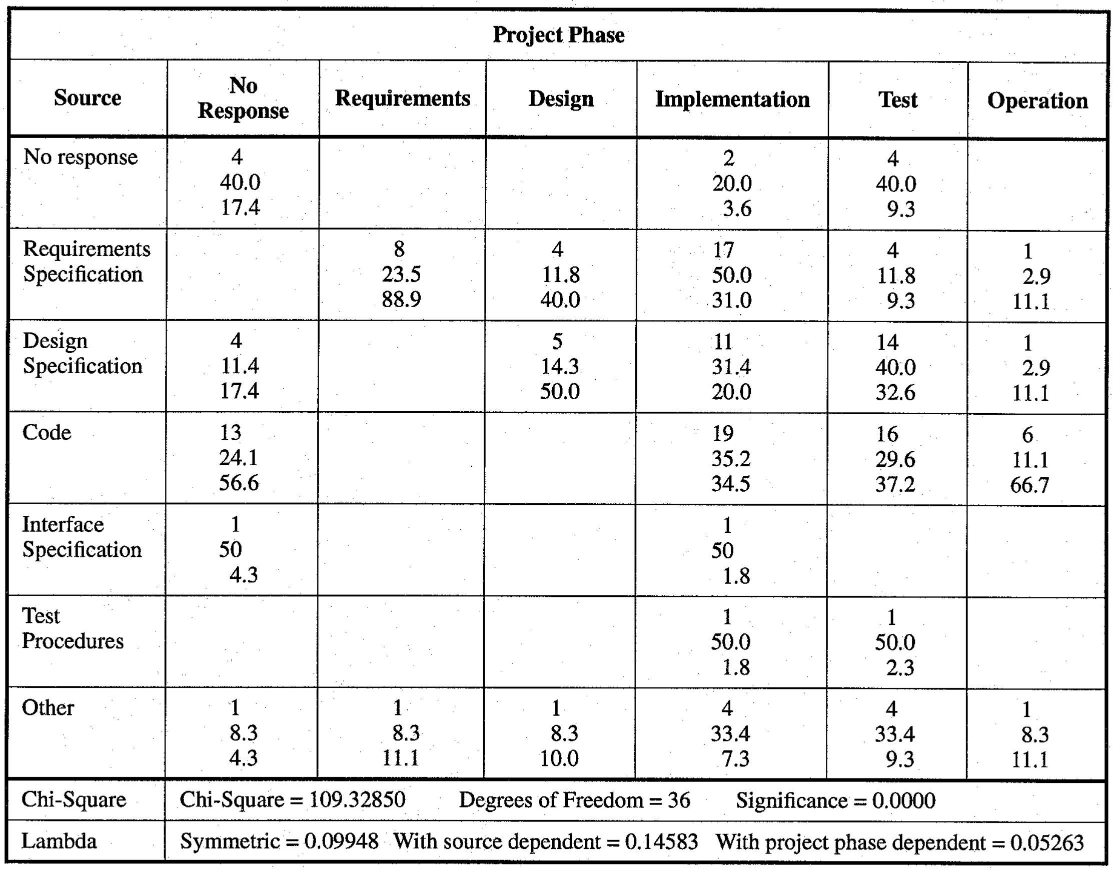
The Chi-Square significance of less than 0.05 indicates that the distribution of frequencies is not uniform; the Logic Problem and Data Handling Problems are significantly greater in number than the other classifications of anomalies. If this were your own data, your challenge now is to identify why that is the case and what you might do about it.

Correlation can be used between two categories to indicate trends or relationships between those categories. For instance, you could compare a project activity with Type to see which Project Activities are most effective at finding what types of anomalies. Or, you could compare the Project Phase to the Source to see how long anomalies remain in the system before they are discovered. Also, the Project Activity could be compared to the Source to verify if certain activities uncover anomalies from certain sources better than others (or Symptom and Type). There are many comparisons that can yield interesting and useful information.

Correlation is an easily-abused statistical method, particularly if you try to use it to identify causation. As long as you use it to identify trends and interesting relationships, this can be a very useful statistic. In an effort to increase the usefulness of this analysis, the Chi-square test can be used to indicate the independence of the categories. If the observed significance level is less than 0.05, the categories are considered dependent; otherwise, they are considered independent. Independence implies no relation between the categories, while dependence implies a relationship between at least one of the classifications in both categories. When there is a dependence, look deeper into your data collection techniques and timing to determine why that dependence might be there and to determine what action, if any, you should take as a result.

Evaluating the measure of association between two categories in a comparison can indicate a predictive relationship between the two categories. This is done with the lambda statistic, where lambda reflects the reduction in error when the classifications of one category are used to predict the classifications of the other. The closer lambda is to zero, the less likely it is that a variable in the analysis is dependent on the other. The closer lambda is to one, the more likely it is that a variable in the analysis is dependent on the other. The lambda calculation results shown in figure 6 evaluate the predictive ability of either variable to the other (symmetric) and then evaluate each variable as the dependent variable, for a total of 3 results. This analysis is useful for identifying dependency trends in your data. It's up to you to figure out what the trends mean.

Figure 6 is an example crosstabulation (comparison) between Source and Project Phase that may come from a statistical analysis package. The numbers in each box represent the count, row percent, and column percent, in that order. This example does not use the entire list of Project Phase or Source described in IEEE Std 1044-1993.



Project Phase						
Source	**No Response**	**Requirements**	**Design**	**Implementation**	**Test**	**Operation**
No response	4 40.0 17.4			2 20.0 3.6	4 40.0 9.3	
Requirements Specification		8 23.5 88.9	4 11.8 40.0	17 50.0 31.0	4 11.8 9.3	1 2.9 11.1
Design Specification	4 11.4 17.4		5 14.3 50.0	11 31.4 20.0	14 40.0 32.6	1 2.9 11.1
Code	13 24.1 56.6			19 35.2 34.5	16 29.6 37.2	6 11.1 66.7
Interface Specification	1 50 4.3			1 50 1.8		
Test Procedures				1 50.0 1.8	1 50.0 2.3	
Other	1 8.3 4.3	1 8.3 11.1	1 8.3 10.0	4 33.4 7.3	4 33.4 9.3	1 8.3 11.1
Chi-Square	Chi-Square = 109.32850 Degrees of Freedom = 36 Significance = 0.0000					
Lambda	Symmetric = 0.09948 With source dependent = 0.14583 With project phase dependent = 0.05263					

Figure 6—Cross-tabulation example using Chi-Square and Lambda

The Chi-Square less than 0.05 indicates that the Source and Project Phase are dependent on each other. The lambda statistic, being very close to zero in all three calculations, indicates no dependence between the Project Phase and the Source of the anomalies.

10.2 Project management

Project management, in the context of this guide, means evaluating the impact of enhancements and anomalies on the project plan. Classification information is used to determine which anomalies will be addressed in the current development release (and which will be deferred to a later release) and to determine whether changing the schedule is necessary in order to address an enhancement or defect. For instance, a high number of high Customer Value defects would indicate that the software is not yet ready for distribution. Tracking the number of anomalies at each Customer Value and/or Priority and/or Severity on a weekly basis can provide a regular assessment.

Comparing the Project Cost, Project Risk, and Impact to the project's quality or reliability can help make a fix or no-fix decision during the last stages of a project's development cycle. These are probably analyses you are already doing; IEEE Std 1044-1993 simply augments or organizes the data for decisions you need to make when managing a project. The specific categories in IEEE Std 1044-1993 which directly relate to project management are Product Status, Customer Value, Severity, Priority, Mission/Safety, Project Cost, Project Risk, Project Quality/Reliability and Project Schedule.

As a brief definition reminder, Customer Value is the value to the customer or user, Severity relates strictly to the anomaly from an objective, engineering point of view, and Priority is how important addressing the change is to the project success. These and the other impact categories give the project manager information that summarizes the outstanding anomalies. You can use the impact categories to assess the importance of addressing the anomalies and to decide whether to adjust the project plan. The concept of prioritizing anomalies is not new. The concept of prioritizing them from several different viewpoints and keeping that data distinct is not as common.

10.3 Process improvement

Process improvement with regards to anomaly tracking refers to using information from a recent release to indicate areas that may be symptomatic of process weaknesses. For instance, a significant number of data handling problems may indicate that a process change in the detailed design methodology is warranted. Or when you discover that most of your anomalies can be attributed to problems with the requirements specification, you might consider changing the process to spend more resources creating, clarifying, and testing the product requirements. Frequently, applying causal analysis can lead to a better process that is less costly and results in fewer anomalies. Another example is a preponderance of logic errors. This fact could be used to help justify, for instance, implementation of a code inspection procedure.

The categories in IEEE Std 1044-1993 which impact process improvement are Type, Source, Actual Cause, Project Phase, and Project Activity. Some software engineering organizations predict the number of anomalies they expect to discover in each project phase. Correlation of the expected anomaly discovery rate with the actual rate (over the entire program life cycle) is then used to indicate the effectiveness of testing or problems with the development processes.

10.4 Product assessment

Management usually wants to have some mechanism for assessing whether or not a software product is ready for release. Although no single measurement should be used as the definitive answer, reviewing the number of known anomalies in a product and relating that to their Severity and Customer Value gives an excellent measurement to use for assessing a product.

Another consideration in assessing the product is the type of anomalies found. If, for instance, most of the anomalies found during the final testing are logic problems, this could point to a more serious underlying problem in the software design that will continue to create problems for customers. Adding a supporting data item of the software module or file in which an anomaly is fixed provides another level of product assessment. This allows you to pinpoint the most problematic modules and evaluating the types of anomalies in those modules. This in turn can lead you to identify what testing, inspection, or other methods will be most effective in cleaning up those modules before release. If you have data on how long certain tests, inspections, or other defect discovery methods take, you can determine more accurately how long it will be before the product is “releasable.”

Tracking the rate of defect discovery and the rate of closure also provide useful product assessment information. In order to calculate a meaningful “rate of defect discovery,” the supporting data item of “hours to detect the anomaly” needs to be recorded so that you know how much work it took to discover the number of bugs. If only 2 hours are spent looking for anomalies then a low “rate of defect discovery” is very misleading.

Another potential use of the anomaly tracking database is in the phase transition acceptance or exit criteria for a product. One criterion might be that all anomalies are recorded in the database. Another criterion might be that all Priority 1 and 2 (high and medium) anomalies are resolved before the phase transition.

11. Relationships to other standards

Several other organizations have produced standards, or standard-like definitions for software anomaly tracking and management.

11.1 Software Engineering Institute Capability Maturity Model

Anomaly information plays a very prominent role in the Software Engineering Institute’s (SEI) Capability Maturity Model [B10] and related works. The CMM requires anomaly reporting and tracking in level 2 although most organizations using the CMM start tracking anomalies in some fashion when they are at level 1. At level 3 anomaly analysis is done. Level 4 also includes the prediction of likely problems or opportunities for improvements. Ultimately at level 5, anomaly prevention is instituted in addition to the reporting, tracking, analysis, and prediction done in the lower levels. Defect prevention involves analyzing anomalies that were encountered in the past and taking specific actions to prevent the occurrence of those types of defects in the future. IEEE Std 1044-1993 satisfies the kinds of anomaly tracking, reporting analysis, and anomaly prevention encouraged by the CMM.

There is a subgroup affiliated with the SEI, the Software Metrics Definition Working Group, that focuses solely on measures for different aspects of software development and quality. Please see their report, [B9] for more information specific to anomaly classification.

11.2 ISO 9000

Section 4.16 of [B6], Control of quality records, specifies “The supplier shall establish and maintain documented procedures for identification, collection, indexing, access, filing, disposition, maintenance, and storage of quality records.” Further, [B5] section 5.10.6, Maintenance records and reports, specifies that information pertaining to maintenance be recorded in predefined formats and retained following defined rules. Section 6.4.1, Product measurement, of [B5] also requires that the records be “used to manage the development and delivery process.” Clearly, organizations seeking to register or retain registration to ISO 9001 are required to track and use anomaly information. IEEE Std 1044-1993 provides a solid foundation for the type and level of information required to be tracked for ISO 9001 registration.

11.3 Department of Defense and military standards

The current DoD software development standard is [B7], though it is in the process of being replaced. A set of software management indicators and the corresponding data is required to be identified. IEEE Std 1044-1993 also satisfies the anomaly tracking and classifying needs described in this standard.

Annex A

(informative)

Sample definitions of categories and classifications in IEEE Std 1044-1993

A.1 Actual cause

Select the actual cause of the defect. This information is used to distinguish the various kinds of anomalies reported

Code	Classification	Description
IV110	Product	
IV111	Hardware	The hardware (computer) on which the product software runs.
IV112	Software	The product software.
IV113	Data	The product database being tested or in use at the time the anomaly occurred. This means that the problem is in the product database or in the database(s) associated with the product software.
IV114	Interface	An interface between two software products or between software and hardware, or between two hardware products.
IV115	Documentation	Any documentation related to the product software. This includes any specifications, deliverable documentation, training materials, production documentation, and maintenance documentation.
IV116	Enhancement	A new requirement or augmentation of an existing capability needed from the product.
IV120	Test System	
IV121	Hardware	The hardware (computer) on which the test software runs.
IV122	Software	The test software, test tools, or test cases.
IV123	Data	The test database in use at the time the anomaly occurred. This means that the problem is in the test database.
IV124	Interface	The anomaly is in the interface between the test system and the product being tested, or between the hardware and test system.
IV125	Documentation	Any documentation related to the test software. This includes any specifications, deliverable documentation, training materials, production documentation, and maintenance documentation.
IV126	Enhancement	A new requirement or augmentation of an existing capability needed from the test system.
IV130	Platform	

Code	Classification	Description
IV131	Hardware	The hardware (computer) on which the product software runs.
IV132	Operating System	The operating system on the hardware on which the product software runs.
IV133	Documentation	Any documentation related to the platform’s hardware or operating system.
IV140	Outside Vendor/Third Party	Any software, database, documentation, or enhancement provided by an outside vendor and used within or in conjunction with the product software.
IV141	Hardware	The hardware (computer) on which the outside vendor or third party software runs.
IV142	Software	The outside vendor’s or third party’s software.
IV143	Data	The third party database being tested or in use at the time the anomaly occurred. This means the problem is in the third party database or in the database(s) associated with the third party software.
IV144	Documentation	The third party database being tested or in use at the time the anomaly occurred. This means the problem is in the third party database or in the database(s) associated with the third party software.
IV145	Enhancement	A new requirement or augmentation of an existing capability needed from the outside vendor or third-party.
IV150	User	Any situation which involves the user’s misunderstanding of procedures, hitting the wrong button, entering the wrong input, and so forth.
IV160	Unknown	Whatever caused the anomaly originally can not be identified or cannot be repeated.

A.2 Corrective action

Identify what long-term corrective action is to be taken to prevent similar problems from occurring again or to facilitate discovery of similar anomalies earlier in the product life cycle.

Code	Classification	Description
AC210	Department Action	Corrective action is applicable within a department or group in the organization.
AC211	Revise Process	A change in some process will prevent this type of anomaly from recurring or to uncover/discover similar anomalies.
AC212	Implement Training	Training of the persons performing a specific activity will reduce the likelihood of repeated anomalies of this nature or to uncover/discover similar anomalies.
AC213	Create/Revise/ Reinforce Standards/ Specifications	A change in the standards or specifications used will prevent this type of anomaly from recurring or to uncover/ discover similar anomalies.
AC214	Reallocate People/ Resources	If more personnel or resources were devoted to a specific job function, this anomaly would be prevented or to uncover/discover similar anomalies.
AC225	Improve/Enforce Audit Activities	More extensive audits or increased enforcement of existing audits would prevent this anomaly from recurring or to uncover/discover similar anomalies.
AC220	Corporate Action	Corrective action is applicable at the corporate level.
AC221	Revise Process	A change in some corporate process will prevent this type of anomaly from recurring.
AC222	Implement Training	Training of the persons performing several different activities organization-wide will reduce the likelihood of repeated anomalies of this nature.
AC223	Create/Revise/Reinforce Audit Activities	A change in corporate standards or specifications used will prevent this type of anomaly form recurring.
AC224	Reallocate People/ Resources	If more personnel or resources were devoted to a specific job function, this anomaly would be prevented.
AC225	Improve/Enforce Audit Activities	More extensive audits or increased enforcement of existing audits at the corporate level would prevent this anomaly from recurring.
AC230	Industry/Government	Corrective action is applicable throughout the industry or throughout government organizations or legislation.

Code	Classification	Description
AC231	Sponsor Research/ Education Programs	Broad research and or educational programs are needed to resolve this type of anomaly.
AC232	Compile/Publish Data	Data on this problem needs to be collected and published for general consumption to prevent this type of anomaly from recurring.
AC233	Create/Revise/Reinforce Standards/Specifications	Industry or government standards or specifications need to be changed to prevent this type of anomaly from recurring.
AC234	Improve/Enforce Audit Activities	Industry or government audit activities need to be improved or more rigorously enforced in order to prevent this type of anomaly from recurring.
AC240	Institutions for Research / Education	The educational system needs to respond to the challenge of improving available education or information on this type of anomaly in order to reduce the likelihood of recurrence.
AC241	Research Problem	Research is needed to identify how to reduce the likelihood of this type of anomaly.
AC242	Develop New Technologies	New technologies are needed to identify how to reduce the likelihood of this type of anomaly
AC243	Test Alternate Approaches	Alternate approaches need to be tested to identify how to reduce the likelihood of this type of anomaly.
AC244	Create/Revise Tests	Tests need to be created or improved to identify how to reduce the likelihood of this type of anomaly.
Ac245	Enforce Educational Standards	Educational standards need to be enforced in order to reduce the likelihood of this type of anomaly.

A.3 Customer Value

See the alternative definitions for Customer Value included in clause 6 of this guide.

A.4 Disposition

Disposition describes whether an anomaly report has been resolved in some manner. It may still be open and passed on to another release, anomaly, or product, or it is closed and considered complete.

Code	Classification	Description
DP110	Closed	The anomaly has been addressed and is considered complete.
DP111	Resolution Implemented	The anomaly resolution has been implemented.
DP112	Not a Problem	The anomaly was not a problem. The software was supposed to behave as it did.
DP113	Not in Scope of Project	The anomaly was not resolvable, or was outside of the planned and executed scope of the project as defined in its specifications.
DP114	Outside Vendor's Problem	The anomaly must be (or has been) fixed by an outside vendor (as it was their problem to fix).
DP115	Duplicate Problem	The anomaly is identical to another reported anomaly, which is being processed. Reference the duplicate anomaly.
DP120	Deferred	The anomaly is deferred to another, specified, release of the product.
DP130	Merged With Another Problem	The anomaly is the same, or will be fixed by the same implementation as the referenced anomaly.
DP140	Referred to Another Project	The anomaly is referred to another project for implementation. The project and new anomaly are referenced.

A.5 Mission/safety

Indicate how severe the impact of the defect is on the mission (intended operation) and/or to the safety of personnel involved in the mission.

Code	Classification	Description
IM410	Urgent	The failure prevents completion of mission or jeopardizes personnel.
IM420	High	The failure adversely affects completion of mission, no workaround solution exists.
IM430	Medium	The failure adversely affects completion of mission, workaround solution exists.
IM440	Low	The failure causes inconvenience or annoyance.
IM450	None	None of the above, or the anomaly concerns an enhancement rather than a failure.

A.6 Priority

Define the importance of addressing an anomaly to your organization. Generally, this will be the same as the Customer Value. However, in some cases, a single customer may feel that an anomaly is of extreme importance, yet another customer might strongly disagree. Priority is a way for your organization to strike a balance when faced with conflicting input from customers, schedules, resources, plans and cost.

Code	Classification	Description
IM210	Urgent	Extremely urgent, resolve immediately.
IM220	High	Resolution required for the next external release.
IM230	Medium	Resolution required for first customer ship (FCS).
IM240	Low	Resolution desired for FCS (automatically promoted to Priority 3 after FCS).
IM250	None	Resolution not required for FCS.

A.7 Product Status

Evaluate the usability of the product with this anomaly present.

Code	Classification	Description
RR610	Unusable	The product cannot be used with this anomaly present.
RR620	Degraded	Some aspects of the product work, but some attributes do not work in its current condition.
RR630	Affected, Use Work-around	The product can be used, but a workaround (from the customer's preferred method of operation) must be used to achieve some capabilities.
RR640	Unaffected	The functioning of the product is unaffected.

A.8 Project Activity

Select the activity that the person reporting the anomaly was involved in when the anomaly appeared or the activity for which the enhancement is needed

Code	Classification	Description
RR110	Analysis	The examination of code, algorithms, documents, or test results to determine their correctness with respect to their intended uses, or to determine their operational characteristics. Some examples of analytical techniques are performance modeling, mathematical analysis, comparison to previous research, and prototyping.
RR120	Review	A meeting during which some part of the product software or documentation is reviewed in order to detect and remedy any deficiencies that could affect its fitness for use or the environmental aspects of the product, process, or service. The review may also identify potential improvements of performance, safety, and economic aspects. The users, customers, or other interested persons are often included for their comments and approval.
RR130	Audit	An independent examination of a work product or set of work products to assess compliance with specifications, standards, contractual agreements, or other criteria.
RR140	Inspection	A static analysis technique that relies on visual examination of development products to detect errors, violations of development standards, and other problems. Types include design and code inspections.
RR150	Code/CompileAssmble	The process of translating a user language program into its relocatable or absolute machine code equivalent.
RR160	Testing	The process of exercising a system or system component by manual or automated means to verify that it satisfies specified requirements or to identify differences between expected and actual results.
RR170	Validation/Qualification Testing	Formal testing, usually conducted by the developer for the customer, to demonstrate that the software meets its specified requirements.
RR180	Support/Operational	The software has been delivered and accepted and is being used for its intended purpose in its intended environment. The software is supported and maintained by the developer or the using agency.
RR190	Walk-through	A static analysis technique in which a designer or programmer leads members of the development team and other interested parties through a segment of documentation or code, and the participants ask questions and make comments about possible errors, violation of development standards, and other problems.

A.9 Project Cost

Project Cost gives a very rough appraisal of the cost required to address a defect or enhancement. This is a qualitative assessment based on the project (and possibly the current phase of development) and on how much addressing the defect or enhancement would affect the project cost. The information provided in this field assists in project planning and change control. Specific dollar amounts can be added to the definitions as appropriate to the organization.

Code	Classification	Description
IM610	High	Fixing this anomaly would be extremely expensive. This change requires a major effort.
IM620	Medium	Fixing this anomaly would be expensive. This enhancement constitutes a moderate effort.
IM630	Low	Fixing this anomaly would have a nominal cost. This change requires little effort.
IM640	None	Fixing this anomaly or implementing this change is included in the budget for this project, so there is no additional cost.

A.10 Project Phase

The software life cycle phase in which the software is when the defect is discovered.

Code	Classification	Description
RR210	Requirements	The period of time in the software life cycle during which the requirements for a software product are defined and documented.
RR211	Concept Evaluation	The period of time during which the product concept is evaluated and justified prior to the development of defined and documented requirements.
RR212	System Requirements	The period of time during which the system requirements are defined and documented.
RR213	Software Requirements	The period of time during which the software requirements are defined and documented.
RR214	Prototype Requirements	The period of time during which the prototype is used to define and document requirements.
RR220	Design	The period of time in the software life cycle during which the designs for architecture, software components, interfaces, and data are created, documented, and verified to satisfy requirements.
RR221	System Design	The period of time during which the collection of components which will make up the system are identified and organized to accomplish the functions specified in the requirements.
RR222	Preliminary Design	The period of time during which the process of analyzing design alternatives and defining the architecture, components, interfaces, and timing and sizing estimates for a system or component occurs
RR223	Detail Design	The period of time during which the process of refining and expanding the preliminary design of a system or component to the extent that the design is sufficiently complete to be implemented occurs.
RR224	Prototype Design	The period of time during which the prototype is designed to satisfy requirements.
RR230	Implementation	The period of time in the software life cycle during which a software product is created from design documentation and debugged.
RR231	Code	The period of time during which the design documentation is expressed as a computer program in a programming language.
RR232	Unit Test	The period of time during which a software product’s individual units or groups of related units are tested.
RR233	Integrate	The period of time during which the software components and/or hardware components are combined into an overall system.

Code	Classification	Description
RR234	Prototype	The period of time during which a preliminary type, form, or instance of a system that serves as a model for later stages or for the final complete version of the system is created from the prototype design documentation and debugged.
RR240	Test	The period of time in the software life cycle during which the components of a software product are evaluated and integrated, and the software product is evaluated to determine whether or not requirements have been satisfied.
RR241	Integration Test	The period of time during which software components, hardware components, or both are combined and tested to evaluate the interaction between them.
RR242	System Test	The period of time during which tests are conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.
RR243	Beta Test	The period of time during which specific customers test and evaluate the usefulness and effectiveness of the system's compliance with its specified requirements in their environment.
RR244	Prototype Test	The period of time during which a prototype is tested and evaluated to determine whether or not requirements have been satisfied and whether or not the product has additional requirements prior to being acceptable in the completed project.
RR245	Acceptance Test	The period of time during which formal testing is conducted to determine whether or not a system satisfies its acceptance criteria or to enable the customer to determine whether or not to accept the system.
RR246	Installation and Checkout	The period of time in the software life cycle during which a software product is integrated into its operational environment and tested in this environment to ensure that it performs as required.
RR250	Operation and Maintenance	The period of time in the software life cycle during which a software product is employed in its operational environment, monitored for satisfactory performance, and modified as necessary to correct problems or to respond to changing requirements.
RR260	Retirement	The period of time in the software life cycle during which support for a software product is terminated. Used for anomalies reported after the product is no longer supported.

A.11 Project Quality/Reliability

Project Quality/Reliability gives you a very rough appraisal of the quality impact on the project if a defect or enhancement is addressed. This is a qualitative assessment based on the project and possibly the current phase of development and on how much addressing the defect or enhancement would affect the project's quality or reliability. The information provided in this field assists in project planning and change control.

Code	Classification	Description
IM810	High	Project quality or reliability will be vastly improved if this defect is fixed or enhancement implemented.
IM820	Medium	Project quality or reliability will be moderately improved if this defect is fixed or enhancement implemented.
IM830	Low	Project quality or reliability will be barely improved if this defect is fixed or enhancement implemented.
IM840	None	Project quality or reliability will not be impacted by the implementation of this defect fix or enhancement.

A.12 Project Risk

Project Risk gives a very rough appraisal of the risk involved in addressing the defect or enhancement. This is a qualitative assessment based on the project and possibly the current phase of development and on how much addressing the defect or enhancement would put the project at risk. The information provided in this field assists in project planning and change control.

Code	Classification	Description
IM710	High	Fixing this anomaly has a high risk of negative impact on the project.
IM720	Medium	Fixing this anomaly has a medium risk of negative impact on the project.
IM730	Low	Fixing this anomaly has little risk of negative impact on the project.
IM740	None	Fixing this anomaly or implementing this enhancement has a negligible risk.

A.13 Project Schedule

This gives you a very rough appraisal of the time required to address a defect or enhancement. This is a qualitative assessment based on the project and possibly the current phase of development and on how much addressing the defect or enhancement would affect the current schedule for the project. The information provided in this field assists in project planning and change control.

Code	Classification	Description
IM510	High	Fixing this anomaly would destroy the current schedule. This enhancement requires a major effort.
IM520	Medium	Fixing this anomaly would push the schedule out significantly. This enhancement constitutes a moderate effort.
IM530	Low	Fixing this anomaly would have little impact on the current schedule, though it might have some. This enhancement requires little effort.
IM540	None	Fixing this anomaly or implementing this enhancement is planned in the current schedule, so there is no impact to the schedule.

A.14 Repeatability

When attempting to reproduce the anomaly, indicate how easy it is to do so.

Code	Classification	Description
RR410	One time occurrence	The anomaly occurred once, but has not been seen more than once.
RR420	Intermittent	The anomaly has occurred more than once, but it is not predictably reproducible.
RR430	Recurring	The anomaly occurs regularly, but with different sets of steps to reproduce it.
RR440	Reproducible	The anomaly can be reproduced with the same well-defined set of steps every time.
RR450	Unknown	No attempt has been made to reproduce the anomaly.

A.15 Resolution

Use to show the decision made by the development or project planning team, that is, whether to implement the change, to defer it, and so forth.

Code	Classification	Description
AC110	Immediate	The modification must be implemented in the current operational release of the product and a special bug-fix release must be prepared for current customers. (Some organizations might attribute a turn-around time to patch rather than wait for specific development cycles.)
AC111	Software Fix	The modification will be through a software change.
AC112	Update Project Documentation	The modification will be through a documentation change.
AC113	Operator Training	The problem will be resolved by training the user.
AC114	Testware Fix	The problem will be resolved in the testware.
AC115	Outside Vendor/Third Party	The problem will be resolved by the appropriate outside vendor or a third-party.
AC120	Eventual	The modification must be implemented in the current development release of the product. (Some organizations might attribute a turn-around time to fix rather than wait for specific development cycles.)
AC121	Software Fix	The modification will be through a software change.
AC122	Update Project Documentation	The modification will be through a documentation change.
AC123	Operator Training	The problem will be resolved by training the user.
AC124	Testware Fix	The problem will be resolved in the testware.
AC125	Outside Vendor/Third Party	The problem will be resolved by the appropriate outside vendor or a third-party.
AC130	Defer	This modification will be implemented in a later, specified version of the software. Note the proposed release version.
AC131	Fix in Later Release	The modification requested will be implemented in a later version of the software. Note the proposed release version.
AC132	Waiver Requested	It will be very difficult to implement the anomaly for this release; a contract waiver is requested so that the software can be released with the anomaly. May fix later in a specified release or patch.
AC140	No Fix	The modification will not be made for this or any subsequent release.
AC141	Not a Problem	The software behaves as intended and as it should remain.
AC142	Waiver Requested	It will be very difficult to implement the change requested; a contract waiver is requested so that the software can be released without the change being made.
AC143	Fix not Justifiable	The problem or enhancement described does not warrant the time and money required to implement it. Or, making this modification will create more problems than it solves.
AC144	Fix not Identifiable	An appropriate fix to solve the problem reported can not be identified.
AC145	Obsolete	The problem has been overcome by hardware, support software, or the current release of the product software or documentation and is no longer a problem.

A.16 Severity

Objectively indicate how severe the impact of the anomaly is on the program operation.

Code	Classification	Description
IM110	Urgent	The failure causes a system crash or unrecoverable data loss or jeopardizes personnel.
IM120	High	The failure causes impairment of critical system functions and no workaround solution exists.
IM130	Medium	The failure causes impairment of critical system functions, though a workaround solution does exist.
IM140	Low	The failure causes inconvenience or annoyance.
IM150	None	None of the above, or the anomaly concerns an enhancement rather than a failure.

A.17 Societal

Societal gives a very rough appraisal of the impact on society (or neighboring communities, at least) if a defect or enhancement is addressed. This category applies to projects where the general public's life, limb, money, or reputation of the organization might be at stake, such as software for a power plant, weapons control, or a communication system. This is a qualitative assessment based on the project and on how much addressing the defect or enhancement would affect society in general. The information provided in this field assists in project planning, change control, and release assessment.

Code	Classification	Description
IM910	High	The likelihood of negative impact to society is high and significantly reduced if this defect is fixed or enhancement is implemented.
IM920	Medium	The likelihood of negative impact to society is moderately high and noticeably reduced if this defect if fixed or enhancement is implemented.
IM930	Low	The likelihood of negative impact to society is low and not significantly reduced if this defect is fixed or enhancement is implemented.
IM940	None	There is no likelihood of negative impact to society and this will not be impacted by the implementation of this defect fix or enhancement.

A.18 Source

Specify the source to which the problem can be traced. This information is used to help identify parts of the development process that need improvement.

Code	Classification	Description
IV210	Specification	Any specification used to define some part of a system's operation that is used in the development of the product software.
IV211	Requirements	A document that specifies the requirements for a system or component. Typically included are functional requirements, performance requirements, interface requirements, design requirements, and development standards.
IV212	Functional	A document that specifies the functions that a system or component must perform. Often part of a requirements specification.
IV213	Preliminary Design	The result of the process of analyzing design alternatives and defining the architecture, components, interfaces, and timing and sizing estimates for a system or component.
IV214	Detailed Design	The result of the process of refining and expanding the preliminary design of a system or component to the extent that the design is sufficiently complete to be implemented.
IV215	Product Design	A document that specifies the overall requirements and functions that a product must provide.
IV216	Interface	A document that specifies the interface characteristics of an existing or planned system or component.
IV217	Data	A document that specifies the data restrictions to be used or the data dictionary.
IV218	Implementation	A document that specifies the how the implementation is to take place. This might include coding standards to be followed, programming languages to be used, or other aspects of the implementation planned.
IV220	Code	Any code, either the product, test system, or outside vendor software.
IV230	Database	Any collection of data used by, or fundamental to, the product software system.
IV240	Manuals or Guides	Any documents intended to provide instructions for how to use the software or that describe the software for customers (not the developers).
IV241	User Guide	Task-oriented documentation describing the way in which a system or component is to be used to obtain desired results.
IV242	Reference Manual	A document that presents the information necessary to employ a particular component to obtain desired results. Typically described are system or component capabilities, limitations, options, permitted inputs, expected outputs, possible error messages, and special instructions. This is distinguished from a system administrator manual in that it is intended for the end users as opposed to those who administer the software and computer system.

Code	Classification	Description
IV243	Product Internals Training Manual	A document that provides the information necessary to develop or modify software for a given computer system.
IV244	System Administrator Manual	A document that provides the information necessary to initiate and operate a system or component. Typically described are procedures for preparation, operating, monitoring, and recovery. This is distinguished from a user guide as being for those who administer a computer system as opposed to those who use it for its intended purpose.
IV245	Installation Guide	A document that provides the information necessary to install a system or component, set initial parameters, and prepare the system or component for operational use.
IV250	Plans and Procedures	Any plans or procedures written and implemented during development, such as test plans, configuration management plans, documentation plans, test cases, etc.
IV251	Test Plan	A document describing the scope, approach, resources, and schedule of intended test activities. It also identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning.
IV252	Test Procedures	A document containing a set of detailed instructions for the setup, execution, and evaluation of results for a given test case.
IV253	Quality Assurance Plan	A document that describes the planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.
IV254	Configuration Management Plan	A document that describes the technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements. This includes handling baselines, configuration identification, configuration control, configuration status accounting, and configuration audits.
IV255	Maintenance Plan	A document that provides the information necessary to service and maintain an operational system or component throughout its life cycle. Typically described are the hardware and software that make up the system or component and procedures for servicing, repairing, extending, or reprogramming it.
IV256	Product Support Plan	A document that describes how customers will be supported in their use of the product, both those involved in the administration of the product and the end-users.
IV260	Reports	Reports generated to summarize various activities during development, such as test reports.
IV261	Test Report	A document that describes the conduct and results of the testing carried out for a system or component.
IV262	Quality Assessment Report	A document written by quality assurance personnel that describes the assessment of the product upon completion of testing.
IV270	Standards/Policies	Standards upon which the design and code must be based. These can be organization policies and standards, industry standards, or government standards.

A.19 Suspected Cause

Select the Cause the submitter suspects for the defect. This information is sometimes helpful during the investigation of the true (actual) cause of the anomaly reported.

Code	Classification	Description
RR310	Product	
RR311	Hardware	The hardware (computer) on which the product software runs.
RR312	Software	The product software.
RR313	Data	The product database being tested or in use at the time the anomaly occurred. This means that the problem is in the product database or in the database(s) associated with the product software.
RR314	Interface	An interface between two software products or between software and hardware, or between two hardware products.
RR315	Documentation	Any documentation related to the product software. This includes any specifications, deliverable documentation, training materials, production documentation, and maintenance documentation.
RR316	Enhancement	A new requirement or augmentation of an existing capability needed of the product.
RR320	Test System	
RR321	Hardware	The hardware (computer) on which the test software runs.
RR322	Software	The test software.
RR323	Data	The test database in use at the time the anomaly occurred. This means that the problem is in the test database.
RR324	Interface	The anomaly is in the interface between the test system and the product being tested, or between the hardware and test system.
RR325	Documentation	Any documentation related to the test software. This includes any specifications, deliverable documentation, training materials, production documentation, and maintenance documentation.
RR326	Enhancement	A new requirement or augmentation of an existing capability needed from the test system.
RR330	Platform	
RR331	Hardware	The hardware (computer and peripherals) on which the product software runs.

Code	Classification	Description
RR332	Operating System	The operating system on the hardware on which the product software runs.
RR333	Documentation	Any documentation related to the platform's hardware or operating system.
RR340	Outside Vendor/Third Party	Any software, database, documentation, or enhancement provided by an outside vendor and used within or in conjunction with the product software.
RR341	Hardware	The hardware (computer) on which the outside vendor or third party software runs.
RR342	Software	The outside vendor's or third party's software.
RR343	Data	The third party database being tested or in use at the time the anomaly occurred. This means that the problem is in the third party database or in the database(s) associated with the third party software.
RR344	Documentation	Any documentation related to the test software. This includes any specifications, deliverable documentation, training materials, production documentation, and maintenance documentation.
RR345	Enhancement	A new requirement or augmentation of an existing capability needed from the outside vendor or third-party.
RR350	User	Any situation which involves the user's misunderstanding of procedures, hitting the wrong button, entering the wrong input, and so forth.
RR360	Unknown	Whatever caused the anomaly originally can not be identified or cannot be repeated.

A.20 Symptom

Choose what that defect or enhancement looks like to the user. The purpose of this classification detail is to document the user's or customer's perception of what may be in the problem.

Code	Classification	Description
RR510	Operating System Crash	The operating system for the computer crashes (panics), requiring a reboot of the operating system.
RR520	Program Hung-Up	The product software stops responding appropriately for an inordinate amount of time and must be restarted.
RR530	Program Crash	The product software unceremoniously dumps the user into the computer's operating system without performing the usual exit sequence.
RR540	Input Problem	The product software does not deal with the input given to it as the user expects it to do. This input problem is more specifically one of the following:
RR541	Correct Input Not Accepted	The product software does not read an input file or accept a manual input correctly.
RR542	Wrong Input Accepted	The product software accepts input that is incorrect and that will produce incorrect results, if any. Appropriate error messages are not displayed.
RR543	Description Incorrect or Missing	The product software does not supply the correct input description as expected from the documentation for the program's operation.
RR544	Parameters Incomplete or Missing	The product software does not specify the units required or all of the options available for a particular input. In other words, it does not support all of the options and arguments specified or wanted.
RR550	Output Problem	The product software does not produce output as the user expects it to do. This output problem is more specifically one of the following:
RR551	Wrong Format	The product software did not write the data into the output file in the correct format, or it did not display the data in the correct format.
RR552	Incorrect Results/Data	The written or displayed data from the product software are not what they should be; the values are unrealistic for the circumstances, etc.
RR553	Incomplete/Missing	The product software did not write or display all of the data expected.
RR554	Spelling/Grammar	The product software output contains incorrect spelling and/or grammar.
RR555	Cosmetic	The product software output contains ugly graphics, unaesthetic information display, etc.
RR560	Failed Required Performance	The product software did not perform as it was expected to do based upon the requirements.
RR570	Perceived Total Product Failure	The product software failed so that the results of the program did not meet one or more perceived critical requirements.
RR580	System Error Message	The computer's operating system displays an error message of some sort.
RR590	Other	Any symptoms not described in another category.

A.21 Type

Specify further what kind of problem or enhancement it is.

Code	Classification	Description
IV310	Logic Problem	Some aspect of logic was omitted or implemented incorrectly in the product.
IV311	Forgotten Cases or Steps	
IV312	Duplicate Logic	
IV313	Extreme Conditions Neglected	
IV314	Unnecessary Function	
IV315	Misinterpretation	
IV316	Missing Condition Test	
IV317	Checking Wrong Variable	
IV318	Iterating Loop Incorrectly	
IV320	Computational Problem	Some aspect of an algorithm was incorrectly coded.
IV321	Equation Insufficient or Incorrect	
IV321.1	Missing Computation	
IV321.2	Operand in Equation Incorrect	
IV321.3	Operator in Equation Incorrect	
IV321.4	Parentheses Used Incorrectly (or not used)	
IV322	Precision Loss	
IV322.1	Rounding or Truncation Fault	
IV322.2	Mixed Modes	
IV323	Sign Convention Fault	
IV330	Interface/Timing Problem	Some aspect of the software or hardware interfaces does not function properly.
IV331	Interrupts Handled Incorrectly	
IV332	I/O Timing Incorrect	
IV332.1	Timing Fault Causes Data Loss	
IV333	Subroutine /Module Mismatch	
IV333.1	Wrong Subroutine Called	
IV333.2	Incorrectly Located Subroutine Call	
IV333.3	Nonexistent Subroutine Called	
IV333.4	Inconsistent Subroutine Arguments	
IV340	Data Handling Problem	Some aspect of data manipulation (except computational manipulation) was handled incorrectly.
IV341	Initialized Data Incorrectly	

Code	Classification	Description
IV342	Accessed or Stored Data Incorrectly	
IV342.1	Flag or Index Set Incorrectly	
IV342.2	Packed/Unpacked Data Incorrectly	
IV342.3	Referenced Wrong Data Variable	
IV342.4	Data Referenced Out of Bounds	
IV343	Scaling or Units of Data Incorrect	
IV344	Dimensioned Data Incorrectly	
IV344.1	Variable Type Incorrect	
IV344.2	Subscripted Variable Incorrectly	
IV345	Scope of Data Incorrect	
IV350	Data Problem	Some aspect of the data used in the software is itself incorrect or missing.
IV351	Sensor Data Incorrect or Missing	
IV352	Operator Data Incorrect or Missing	
IV353	Embedded Data in Tables Incorrect or Missing	
IV354	External Data Incorrect or Missing	
IV355	Output Data Incorrect or Missing	
IV356	Input Data Incorrect or Missing	
IV360	Documentation Problem	Problems specific to documents rather than to code. They include the following:
IV361	Ambiguous Statement	The statement can be taken to mean several different things.
IV362	Incomplete Item	The statement or description does not seem to consider all aspects of the situation it attempts to describe.
IV363	Incorrect Item	The statement or description is incorrect.
IV364	Missing Item	The statement or description that must be included in the document is missing.
IV365	Conflicting Items	Two or more statements or descriptions conflict or contradict each other.
IV366	Confusing Items	The statement or description confuses the reader.
IV367	Redundant Items	The statement repeats another statement and detracts from clarity rather than adding to it.
IV368	Illogical Item	The statement does not make sense in reference to other statements within the same document or other documents to which it refers.
IV369	Non-verifiable Item	The statement (usually a requirement or functional description) cannot be verified by any reasonable testing methods.
IV370	Unachievable Item	The statement (usually a requirement or functional description) cannot be true in the reasonable lifetime of the product.
IV380	Document Quality Problem	There is a problem with the document as a whole, including the following:

Code	Classification	Description
IV381	Applicable Standards Not Met	Internal or industry standards for the document in question are not met in accordance with organization policy.
IV382	Not Traceable	Items cannot be traced to the appropriate previous or subsequent documents.
IV383	Not Current	The document is not in accordance with the product software or subsequent document
IV384	Incomplete	The document is missing an important section or segment.
IV385	Inconsistencies	The document contains information that is inconsistent with information in another document.
IV390	Enhancement	An enhancement is a suggestion that changes the requirements for an existing product or that changes the code in an existing product to increase the code's efficiency or maintainability.
IV391	Change in Program Requirements	Add a new capability, remove a capability that is now unnecessary, or update an existing capability to meet more user needs.
IV391.1	Add New Capability	
IV391.2	Remove Unnecessary Capability	
IV391.3	Update Current Capability	
IV392	Improve Comments	Expand or change the comments within the program so that they reflect more clearly what the software does.
IV393	Improve Code Efficiency	Improve the efficiency of the code (for example, the execution speed) or increase available memory. Improving the efficiency will not change the external capabilities of the product.
IV394	Implement Editorial Changes	Change the organization of the code so that it is more understandable to those who maintain the code. Editorial changes do not affect the external capabilities of the product.
IV395	Improve Usability	Change a section of the code so that it is more understandable and helpful to the user of the product.
IV396	Software Fix of a Hardware Problem	The hardware is unable to perform certain actions, so the software is altered to work around the hardware limitations.
IV397	Other Enhancement	An enhancement other than one of the types listed above.
IV398	Failure Caused by Previous Fix	The failure was caused by fixing another problem or adding another modification.
IV399	Performance Problem	The speed of the product or a product component is unacceptable.
IV400	Interoperability Problem	The product or a product component is not compatible with other software products or product components.
IV401	Standards Conformance Problem	The product or product component does not conform to a standard, where conformance to a particular standard is specified in the requirements.
IV402	Other	None of the above.

Annex B

(informative)

Bibliography

[B1] DoD-STD-2167A, Defense Systems Software Development, 1988.[1]

[B2] DoDI-5000.2, Defense Acquisition Management Policies and Procedures, 1991.

[B3] IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology (ANSI).

[B4] IEEE Std 1074-1991, IEEE Standard for Developing Software Life Cycle Processes.

[B5] ISO 9000-3:1991, Quality management and quality assurance standards—Part 3: Guidelines for the application of ISO 9001 to the development, supply and maintenance of software. International Organization for Standardization, Geneva, Switzerland, 1991.[2]

[B6] ISO 9001:1994, Quality systems - Model for quality assurance in design, development, production, installation and servicing. International Organization for Standardization, Geneva, Switzerland, 1994.

[B7] MIL-STD-498, Software Development and Documentation, 1994.[3]

[B8] MIL-STD-1512B, Technical Reviews and Audits, 1985.

[B9] SEI-92-TR-022, Software Quality Measurement: A Framework for Counting Problems and Defects, Software Engineering Institute, 1992.

[B10] SEI-93-TR-024, Software Engineering Institute's Capability Maturity Model for Software (CMM v1.1), 1993.

[1] DoD publications are available from Customer Service, Defense Printing Service, 700 Robbins Avenue, Bldg. 4, Philadelphia, PA 19111-5094

[2] ISO publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse. ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

[3] MIL publications are available from Customer Service, Defense Printing Service, 700 Robbins Ave., Bldg. 4D, Philadelphia, PA 19111-5094.