IEEE Standard Classification for Software Anomalies

Sponsor

Software Engineering Standards Committee
of the
IEEE Computer Society

Approved December 2, 1993

IEEE Standards Board

Abstract: A uniform approach to the classification of anomalies found in software and its documentation is provided. The processing of anomalies discovered during any software life cycle phase are described, and comprehensive lists of software anomaly classifications and related data items that are helpful to identify and track anomalies are provided. This standard is not intended to define procedural or format requirements for using the classification scheme. It does identify some classification measures and does not attempt to define all the data supporting the analysis of an anomaly.
Keywords: anomaly, category, classification, classification process, supporting data item

---

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

Print: ISBN 1-55937-708-9, SH94399
PDF: ISBN 0-7381-0406-X, SS94399

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

Introduction

(This introduction is not a part of IEEE Std 1044-1993, IEEE Standard Classification for Software Anomalies.)

This standard provides a uniform approach to the classification of anomalies found in software and its documentation. It describes the processing of anomalies discovered during any software life cycle phase, and it provides comprehensive lists of software anomaly classifications and related data items that are helpful to identify and track anomalies. The minimum set of classifications deemed necessary for a complete data set are indicated as mandatory. More detailed classifications are provided for those projects that require more rigor. These lower levels of detail are shown as optional. Application of less than the mandatory set of classifications is not recommended as this may result in insufficient detail for meaningful data collection and analysis.

Some guidelines on how to apply the standard are provided in 4.1. In addition, annex A provides an example anomaly reporting mechanism that describes how to apply the classification scheme. The classification scheme in this standard considers the environment and activity in which the anomaly occurred, the symptoms of the anomaly, the software or system cause of the anomaly, whether the anomaly is a problem or an enhancement request, where the anomaly originated (by phase and document), the resolution and disposition of the anomaly, the impact of several aspects of the anomaly, and the appropriate corrective action.

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

At the time this standard was completed, the Classification Standards Working Group had the following membership:

Richard Evans, *Chair* **Cynthia Brehmer,** *Co-chair*
Jaya Carl, *Secretary*

John B. Bowen
Lynn K. Broderson
Linda Clemens
Richard J. Gale
Myron Lipow
Mary Mikhail
Patricia Pratt
David Schucker
David Simkins
Vijaya Srivastava
Theodore Sullivan
Greg Ward

The following persons were on the balloting committee:

A. Frank Ackerman
Eleanor Antreassian
Bruce M. Bakken
Jack N. Barnard
Boris Beizer
H. R. Berlack
Barry Boehm
Kathleen L. Briggs
Christian Brunelle
Fletcher J. Buckley
Elliot J. Chikofsky
Jung K. Chung
Francois Coallier
Stewart Crawford
Patricia W. Daggett
James Dobbins
David Doty
William P. Dupras
Robert E. Dwyer
Kenneth Dymond
Caroline L. Evans
James R. Evans
John W. Fendrich
Glenn S. Fields
Violet Foldes
Kenneth A. Foster
Richard Fries
David Gelperin
Anne Geraci
Jean A. Gilmore
Shirley A. Gloss-Soler
Paul Grizenko
L. M. Gunther
Virl Haas
Peter J. Harvey
John Horch
Laurel V. Kaleda
Adi Kasad
R. A. Kessler
Tom Kurihara
Lak-Ming Lam
John B. Lane
Robert A. Lane
F. C. Lim
Ben Livson
Donald Lundry
Austin J. Maher
Kartik C. Majumdar
David M. Marks
Philip C. Marriott
Darrell Marsh
Glen A. Meldrum
Edward F. Miller
Celia H. Modell
Gary D. Moorhead
Gene T. Morun
Robert C. Natale
Geraldine Neidhart
Dennis Nickle
Michael T. Perkins
William E. Perry
Ron Pfaff
Donald J. Pfeiffer
John N. Postak
Robert M. Poston
Jock Rader
Arthur S. Robinson
Frances A. Ruhlman
Margaret Rumley
Norman Schneidewind
Leonard W. Seagren
Harlan K. Seyfer
Anthony F. J. Sgarlatti
Robert W. Shillato
Jacob Slonim
Wayne Smith
Terrence L. Tillmanns
David B. Turner
William Stephen Turner
Ralph Wachter
Dolores R. Wallace
John P. Walter
Andrew H. Weigel
Bill Wong
Dennis L. Wood

When the IEEE Standards Board approved this standard on December 2, 1993, it had the following membership:

Wallace S. Read, *Chair* **Donald C. Loughry,** *Vice Chair*
Andrew G. Salem, *Secretary*

Gilles A. Baril
José A. Berrios de la Paz
Clyde R. Camp
Donald C. Fleckenstein
Jay Forster*
David F. Franklin
Ramiro Garcia
Donald N. Heirman
Jim Isaak
Ben C. Johnson
Walter J. Karplus
Lorraine C. Kevra
E. G. "Al" Kiener
Ivor N. Knight
Joseph L. Koepfinger*
D. N. "Jim" Logothetis
Don T. Michael*
Marco W. Migliaro
L. John Rankine
Arthur K. Reilly
Ronald H. Reimer
Gary S. Robinson
Leonard L. Tripp
Donald W. Zipse

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
James Beall
Richard B. Engelman
David E. Soffrin
Stanley I. Warshaw

Rachel A. Meisel
IEEE Standards Project Editor

Contents

IEEE Standard Classification for Software Anomalies

1. Overview

1.1 Background

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

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

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

1.2 Scope

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

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

a) Definitions of terms not provided in IEEE Std 610.12-1990[1]
b) A basic process (sequence of steps) for classifying and establishing categories of anomalies relating to software products and providing related data and information
c) A standard set of categories and classifications, either mandatory or optional
d) A list of supporting data items

This standard identifies those essential (mandatory) categories needed to establish a common definition. The categories provide a common terminology and concepts to communicate among projects, software development environments, and personnel. It is assumed that all applicable classifications within a mandatory category shall be used for compliance to this standard. This standard also provides categories for additional detail that are not necessarily essential to all projects. These additional categories are identified as optional. This standard is not intended to define procedural or format requirements for using the classification scheme. This standard identifies some classification measures and does not attempt to define all the data supporting the analysis of an anomaly.

The classifications are layered with greater detail in each subsequent layer. For instance, in table 1b the *Project phase* category has two levels of classifications, and in table 3c the *Type* category has three levels of classifications. Each project using this classification scheme may choose the level of classification appropriate to that project. Only the first level of classifications is mandatory for each mandatory category.

The classifications in this standard are not exhaustive. Some projects will have requirements that are unique. It is assumed that in those situations the project members will create project-specific categories and/or classifications in addition to the classifications stated in this standard. Anomaly reports generated using the classifications stated in this standard facilitate the generation of statistics and information for trend analysis.

2. References

This standard shall be used with the following publications:

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

Brehmer, C. L., Carl, J. R., "Incorporating IEEE Std 1044 into Your Anomaly Tracking Process," *Crosstalk*, no. 40, pp. 9–16, Jan. 1993.

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

3. Definitions

Definitions of terms used in this standard are consistent with IEEE Std 610.12-1990, with the few exceptions noted below. Note that the term *anomaly* as defined in this standard expands upon the limited definition included in IEEE Std 610.12-1990. This standard's definition of anomaly includes deviations from the user's perceptions or experiences.

3.1 anomaly: Any condition that deviates from expectations 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.

3.3 classification: A choice within a category.

3.4 classification process: The classification process is a series of activities, starting with the recognition of an anomaly through to its closure. The process is divided into four sequential steps interspersed with three administrative activities. The sequential steps are as follows:

a) Step 1: Recognition
b) Step 2: Investigation
c) Step 3: Action
d) Step 4: Disposition

The three administrative activities applied to each sequential step are as follows:

a) Recording
b) Classifying
c) Identifying impact

3.5 mandatory category: A category that is essential to establish a common definition and to provide common terminology and concepts for communication among projects, business environments, and personnel.

3.6 optional category: A category that provides additional details that are not essential but may be useful in particular situations.

3.7 supporting data item: Data used to describe an anomaly and the environment in which it was encountered.

4. Classification standard

This standard refers to the anomaly classification process shown in figure 1. The standard classification scheme is described in 4.2. The standard categories and classifications are shown in the odd-numbered tables. The supporting data items are shown in the even-numbered tables.

Because the classification scheme depends on an understanding of the basic classification process, the classification process will be discussed first.

4.1 Classification process

The classification process is a series of activities, starting with the recognition of an anomaly through to its closure. The process is divided into four sequential steps interspersed with three administrative activities. The sequential steps are as follows:

a) Step 1: Recognition
b) Step 2: Investigation
c) Step 3: Action
d) Step 4: Disposition

The three administrative activities applied to each sequential step are as follows:

a) Recording
b) Classifying
c) Identifying impact

Each of the steps and administrative activities is discussed below. Figure 1 illustrates the process and interrelationships between the activities. It is anticipated that in the processing of any one anomaly these steps may be repeated several times. It is also expected that individual organizations implementing this standard will use procedures that accomplish the standard's objective but may call for additional detail, a different sequence of steps, and/or may specify organizational involvement.

4.1.1 Recognition

The recognition step occurs when an anomaly is found. Recognition of an anomaly may be made by anyone involved in the development, evaluation, use, support, or maintenance of software, regardless of where in the software life cycle the anomaly is discovered.

a) *Recording the recognition*. When an anomaly is recognized, supporting data items are recorded to identify the anomaly and the environment in which it occurred. The applicable supporting data items indicated in table 2 are recorded as part of the recognition process.
b) *Classifying the recognition*. Certain important attributes of an anomaly are observed and shall be identified during the recognition process. Tables 1a–1f contain the categories of information and the scheme by which these attributes shall be classified.
c) *Identifying impact*. The person discovering the anomaly shall identify his or her perception of the impact of the anomaly. Tables 7a–7i list the categories used for impact identification.

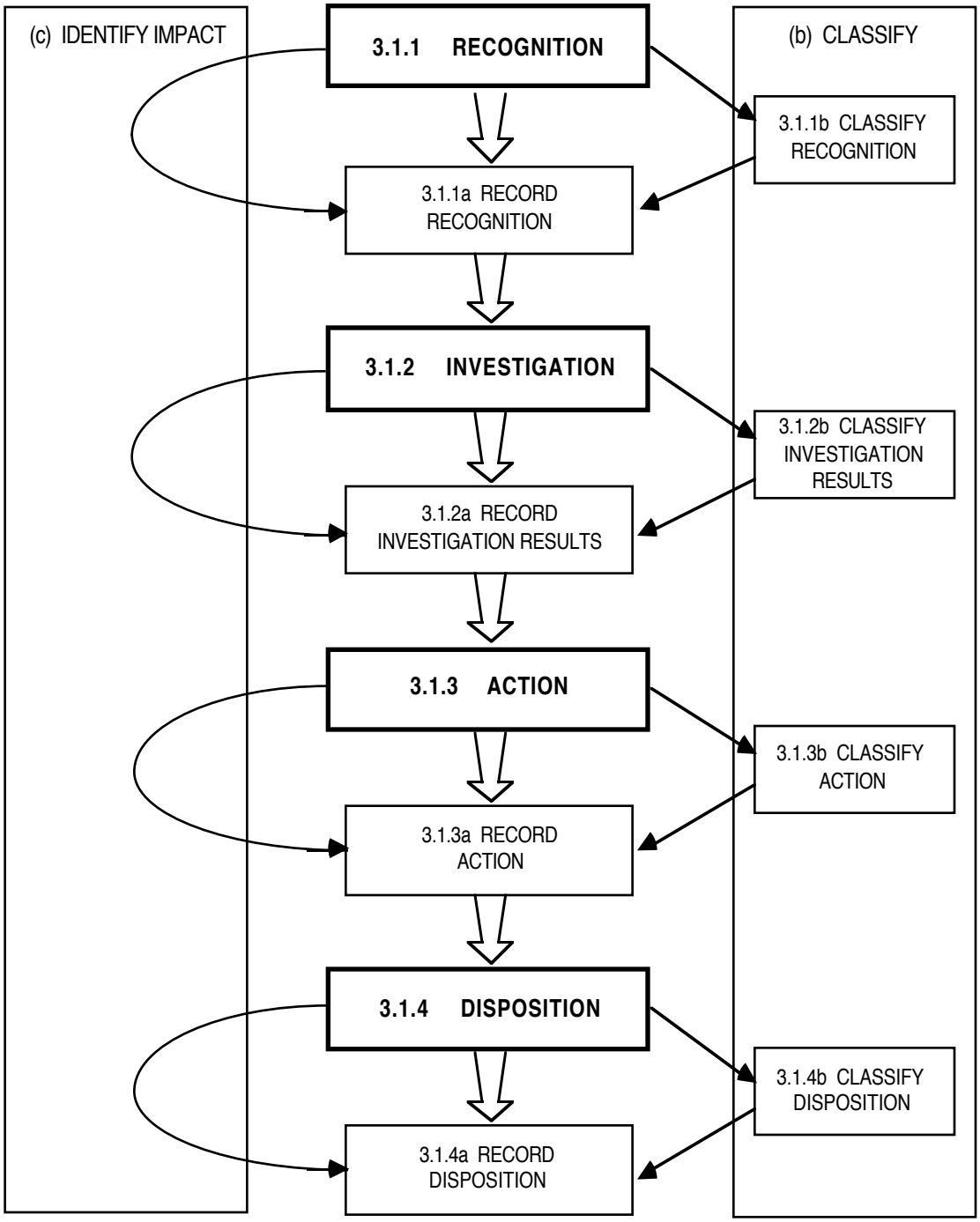

Figure 1—Classification process flow chart

4.1.2 Investigation

Following the recognition step, each anomaly shall be investigated. This investigation shall be of sufficient depth either to identify all known related issues and propose solutions or to indicate that the anomaly requires no action.

a) *Recording the investigation*. The supporting data items indicated as in table 4 are recorded as part of the investigation process. Any new information related to the recognition of the anomaly is updated.
b) *Classifying the investigation*. During the investigation, entries shall be provided for all categories identified as mandatory in tables 3a–3c. In addition, classification entries made during the recognition step shall be reviewed and corrected where appropriate.
c) *Identifying impact*. Previous impact classifications shall be reviewed and updated based on the results of the analysis. Tables 7a–7i list the categories used for impact identification.

4.1.3 Action

A plan of action shall be established based on the results of the investigation. The action includes all activities necessary to resolve the immediate anomaly and those activities required to revise processes, policies, or other conditions necessary to prevent the occurrence of similar anomalies.

a) *Recording the action*. The supporting data items necessary to record the decisions on how to handle the anomaly are in table 6. The applicable supporting data items in this section are recorded.
b) *Classifying the action*. Once the appropriate action or actions are determined, the mandatory categories in tables 5a and 5b shall be classified. Note that the action taken to resolve an anomaly is related to the disposition of the anomaly.
c) *Identifying impact*. Classifications chosen for categories determined in previous steps shall be reviewed and updated as required. Tables 7a–7i list the categories used for impact identification.

4.1.4 Disposition

Following the completion of either all required resolution actions or at least identification of long-term corrective action, each anomaly shall be disposed of by:

a) *Recording the disposition*. The applicable supporting data items in table 10 are recorded for each anomaly.
b) *Classifying the disposition*. Anomalies shall be completed using one of the disposition classifications shown in table 9.
c) *Identifying impact*. Previously recorded impact categories shall be reviewed and updated. All classifications shown in tables 7a–7i as mandatory shall be completed.

4.2 Standard classification scheme

The following subclauses describe how the classification scheme is used in the odd-numbered tables. The classification scheme is structured around the process steps described in figure 1, including recognition, investigation, action, recording, classifying, impact identification, and disposition.

Throughout the classification scheme, there are some occasions when a reference is required. Also, there are instances where more than one classification may be applicable within a category, such as *Type*. In these cases, more than one classification can be indicated, though only one classification per category is normally necessary.

4.2.1 Classification codes

The classification codes, primarily listed in the third column of the odd-numbered tables, are an alphanumeric code (useful for data entry purposes) for each of the categories and classifications shown in the first and fourth columns of the tables, respectively.

Each of the steps or activities being classified is assigned a two-character alpha prefix: RR for Recognition step, IV for Investigation step, AC for Action step, IM for Impact Identification activity, and DP for Disposition step.

The prefix is followed by three digits, identifying the categories and classifications. Where further definition is needed, a decimal number is assigned. For instance, in table 3c classification code IV321.1 first guides the user to the Investigation step (IV). Second, the category the classification belongs to is *Type* (IV300). The *Type* of anomaly is identified as a *Computational problem* (IV320), further identified as an *Equation insufficient or incorrect* (IV321), and, more specifically, as a *Missing computation* (IV321.1).

The supporting data items are necessary for recognition, investigation, action, and disposition of an anomaly. Since each project offers a unique set of relevant supporting data items, this standard does not mandate specific supporting data items. Suggested supporting data items pertinent to each process step are in the even-numbered tables.

4.2.2 Compliance required

In the odd-numbered tables is a column marked *Compliance required*. This identifies the importance (mandatory or optional) of the category for compliance to this standard. It is assumed that all applicable classifications within a mandatory category shall be used for compliance to this standard. Only the first level of classifications is mandatory for each mandatory category. Additional classifications may be added when implementing this standard to provide the analyst with needed accuracy.

4.2.3 Recognition

During the recognition process members of a project shall identify the *Project activity* taking place at the time the anomaly was encountered. The *Project activity* category, which is mandatory, includes identifiable activities that are employed during the development and release of a software product. These activities shall be performed during more than one of the project phases. The *Project phase* category is also mandatory and shall be identified and tailored to the project life cycle being employed when implementing this standard. At the current time a consensus has not developed as to which project phases compose the software life cycle. This standard provides software life cycle phase classifications that can be mapped to many specific software life cycles, including the one in IEEE Std 610.12-1990. The *Suspected cause* category provides the anomaly discoverer the opportunity to speculate on the cause of the anomaly. This category is optional. The *Repeatability* category of the anomaly is optional. Classifying the *Symptom* of the anomaly is mandatory. The *Product status* identifies the usability of the product with the anomaly; it is optional.

Recognition categories and classification codes all have an RR prefix, and are shown in tables 1a–1f.

Table 1a—Recognition classification scheme—Project activity

Category	Compliance required	Code	Classification
Project activity RR100	*Mandatory*	RR110	Analysis
	Mandatory	RR120	Review
	Mandatory	RR130	Audit
	Mandatory	RR140	Inspection
	Mandatory	RR150	Code/Compile/Assemble
	Mandatory	RR160	Testing
	Mandatory	RR170	Validation/Qualification testing
	Mandatory	RR180	Support/Operational
	Mandatory	RR190	Walk-through

Table 1b—Recognition classification scheme—Project phase

Category	Compliance required	Code	Classification
Project phase RR200	*Mandatory*	RR210	Requirements
		RR211	Concept evaluation
		RR212	System requirements
		RR213	Software requirements
		RR214	Prototype requirements
	Mandatory	RR220	Design
		RR221	System design
		RR222	Preliminary design
		RR223	Detail design
		RR224	Prototype design
	Mandatory	RR230	Implementation
		RR231	Code
		RR232	Unit test
		RR233	Integrate
		RR234	Prototype
	Mandatory	RR240	Test
		RR241	Integration test
		RR242	System test
		RR243	Beta test
		RR244	Prototype test
		RR245	Acceptance test
		RR246	Installation and checkout
	Mandatory	RR250	Operation and maintenance
	Mandatory	RR260	Retirement

Table 1c—Recognition classification scheme—Suspected cause

Category	Compliance required	Code	Classification
Suspected cause RR300	*Optional*	RR310	Product
		RR311	Hardware
		RR312	Software
		RR313	Data
		RR314	Interface
		RR315	Documentation
		RR316	Enhancement (Perceived inadequacies)
		RR320	Test system
		RR321	Hardware
		RR322	Software
		RR323	Data
		RR324	Interface
		RR325	Documentation
		RR326	Enhancement (Perceived inadequacies)
		RR330	Platform
		RR331	Hardware
		RR332	Operating system
		RR333	Documentation
		RR340	Outside vendor/Third party
		RR341	Hardware
		RR342	Software
		RR343	Data
		RR344	Documentation
		RR345	Enhancement (Perceived inadequacies)
		RR350	User
		RR360	Unknown

Table 1d—Recognition classification scheme—Repeatability

Category	Compliance required	Code	Classification
Repeatability RR400	*Optional*	RR410	One time occurrence
		RR420	Intermittent
		RR430	Recurring
		RR440	Reproducible
		RR450	Unknown

Table 1e—Recognition classification scheme—System

Category	Compliance required	Code	Classification
Symptom RR500	*Mandatory*	RR510	Operating system crash
	Mandatory	RR520	Program hang-up
	Mandatory	RR530	Program crash
	Mandatory	RR540	Input problem
		RR541	Correct input not accepted
		RR542	Wrong input accepted
		RR543	Description incorrect or missing
		RR544	Parameters incomplete or missing
	Mandatory	RR550	Output problem
		RR551	Wrong format
		RR552	Incorrect result/data
		RR553	Incomplete/Missing
		RR554	Spelling/Grammar
		RR555	Cosmetic
	Mandatory	RR560	Failed required performance
	Mandatory	RR570	Perceived total product failure
	Mandatory	RR580	System error message
	Mandatory	RR590	Other

Table 1f—Recognition classification scheme—Product status

Category	Compliance required	Code	Classification
Product status RR600	*Optional*	RR610	Unusable
		RR620	Degraded
		RR630	Affected, use workaround
		RR640	Unaffected

4.2.3.1 Recognition supporting data items

It is generally necessary to collect considerable information in addition to classifiable information during the recognition process of an anomaly. Table 2 contains a list of important, but optional, supporting data items. Note that this list is not exhaustive.

Table 2—Recognition supporting data items

Environment		Originator	Anomaly definition	Time	Vendor
Product hardware	Name	Name	Anomaly description	Operate time	Company
	Revision	Date anomaly occurred	Customer irritability factor	Wall clock time	Contact
	ID number/model	Code or functional area	Triggering event	System time	Vendor's ID number
	Serial number	E-mail address	Work around or patch (means to avoid)	CPU time	Expected resolution
Product software	Name	Address	Proposed fix		Expected resolution date
	Version/revision level	Phone number	Documentation used (attached as applicable)		
Database	Name	Company identification			
	Version/revision level	Distribution (activities ID #)			
Test support software	Name				
	Version/revision level				
Platform	Name				
	ID number/model				
	Serial number				
	Version/revision level				
	Operating system version				
Firmware	Name				
	ID number/model				
	Serial number				
	Version/revision level				
Other	Monitor				
	Network				
	Peripherals				

4.2.4 Investigation

The next process step is investigation of the anomaly. The *Actual cause*, *Source*, and the *Type* of the anomaly shall be identified. For instance, if an anomaly report were written because the testing procedure was incorrect, then the *Actual cause* would be IV125 *Test system documentation* and the *Source* would be IV252 *Test procedures*.

The *Type* of anomaly identifies whether the anomaly is in the software documentation, an enhancement, or some aspect of the code itself, such as its logic, interfaces, data handling, data, or computations. There are several levels of the classifications in the *Type* category so that the degree or depth of classification may be chosen for each project beyond the mandatory first level.

Extending the testing procedure incorrect example mentioned in the first paragraph of this subclause, the *Type* could be either IV363, incorrect item in document, or IV316, missing condition test in logic. If test procedures are mostly written in text as opposed to code, the document problem assignation is more appropriate. If test procedures are code or pseudocode, the logic problem classification is better.

Investigation categories and classification codes have an IV prefix, and are listed in tables 3a–3c.

Table 3a—Investigation classification scheme—Actual cause

Category	Compliance required	Code	Classification
Actual cause IV100	*Mandatory*	IV110	Product
		IV111	Hardware
		IV112	Software
		IV113	Data
		IV114	Interface
		IV115	Documentation
		IV116	Enhancement (perceived inadequacies)
	Mandatory	IV120	Test system
		IV121	Hardware
		IV122	Software
		IV123	Data
		IV124	Interface
		IV125	Documentation
		IV126	Enhancement (perceived inadequacies)
	Mandatory	IV130	Platform
		IV131	Hardware
		IV132	Operating system
		IV133	Documentation
	Mandatory	IV140	Outside vendor/Third party
		IV141	Hardware
		IV142	Software
		IV143	Data
		IV144	Documentation
		IV145	Enhancement (perceived inadequacies)
	Mandatory	IV150	User
	Mandatory	IV160	Unknown

Table 3b—Investigation classification scheme—Source

Category	Compliance required	Code	Classification
Source IV200	*Mandatory*	IV210	Specification
		IV211	Requirements
		IV212	Functional
		IV213	Preliminary design
		IV214	Detailed design
		IV215	Product design
		IV216	Interface
		IV217	Data
		IV218	Implementation
	Mandatory	IV220	Code
	Mandatory	IV230	Database
	Mandatory	IV240	Manual and guides
		IV241	User guide
		IV242	Reference manual
		IV243	Product internals training manual
		IV244	System administrator manual
		IV245	Installation guide
	Mandatory	IV250	Plans and procedures
		IV251	Test plan
		IV252	Test procedures
		IV253	Quality assurance plan
		IV254	Configuration management plan
		IV255	Maintenance plan
		IV256	Product support plan
	Mandatory	IV260	Reports
		IV261	Test report
		IV262	Quality assessment report
	Mandatory	IV270	Standards/Policies

Table 3c—Investigation classification scheme—Type

Category	Compliance required	Code	Classification
Type IV300	*Mandatory*	IV310	Logic problem
		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
	Mandatory	IV320	Computation problem
		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
		IV322	Precision loss
		IV322.1	Rounding or truncation fault
		IV322.2	Mixed modes
		IV323	Sign convention fault
	Mandatory	IV330	Interface/timing problem
		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
	Mandatory	IV340	Data handling problem
		IV341	Initialized data incorrectly
		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

Table 3c—Investigation classification scheme—Type (Continued)

Category	Compliance required	Code	Classification
Type IV300		IV344.2	Subscripted variable incorrectly
		IV345	Scope of data incorrect
	Mandatory	IV350	Data problem
		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
	Mandatory	IV360	Documentation problem
		IV361	Ambiguous statement
		IV362	Incomplete item
		IV363	Incorrect item
		IV364	Missing item
		IV365	Conflicting items
		IV366	Redundant items
		IV367	Confusing item
		IV368	Illogical item
		IV369	Nonverifiable item
	Mandatory	IV370	Unachievable item
	Mandatory	IV380	Document quality problem
		IV381	Application standards not met
		IV382	Not traceable
		IV383	Not current
		IV384	Inconsistencies
		IV385	Incomplete
	Mandatory	IV390	Enhancement
		IV391	Change in program requirements
		IV391.1	Add new capability
		IV391.2	Remove unnecessary capability
		IV391.3	Update current capability
		IV392	Improve comments
		IV393	Improve code efficiency
		IV394	Implement editorial changes
		IV395	Improve usability
		IV396	Software fix of a hardware problem
		IV397	Other enhancement

Table 3c—Investigation classification scheme—Type (Concluded)

Category	Compliance required	Code	Classification
IV390/400		IV398	Failure caused by a previous fix
		IV399	Performance problem
		IV400	Interoperability problem
		IV401	Standards conformance problem
		IV402	Other problem

4.2.4.1 Investigation supporting data items

The investigation process, in this context, includes the verification that the anomaly can be reproduced or that it exists, the possible options for dealing with the anomaly, and an analysis of what it would take to implement the possible changes. Table 4 lists some supporting data items that may need documenting during this process. Again, no attempt has been made to make this list exhaustive.

Table 4—Investigation supporting data items

Acknowledgment	Verification
Date received	Source of anomaly
Report number assigned	Data from recognition process
Investigator	
Name	
Code or functional area	
E-mail address	
Phone number	
Estimated start date of investigation	
Estimated complete date of investigation	
Actual start date of investigation	
Actual complete date of investigation	
Person hours	
Date receipt acknowledgment	
Documents used in investigation	
Name	
ID number	
Revision	

4.2.5 Action

Taking action is the next step in the classification process described in figure 1. The minimum classification action required shall be identifying the *Resolution* of the anomaly. The *Resolution* defines how the analyst should choose to deal with the anomaly. An immediate resolution means that the change is implemented immediately, outside of the regular release cycle. The eventual resolution means that the anomaly will be

addressed within the regular release cycle, implemented in the release currently in development. The deferred resolution means that the anomaly may be addressed in a future release, preferably one that is in the planning stages.

The *Corrective action* category is optional because not all anomalies considered will require individual corrective actions. When appropriate, such as for an anomaly that can be traced to some procedure and that could be avoided with different procedures, the recommended corrective action should be noted.

Action categories and classification codes all have an AC prefix, and are listed in tables 5a and 5b.

Table 5a—Action classification scheme—Resolution

Category	Compliance required	Code	Classification
Resolution AC100	*Mandatory*	AC110	Immediate
		AC111	Software fix
		AC112	Update project documentation
		AC113	Operator training
		AC114	Testware fix
		AC115	Outside vendor/Third party
	Mandatory	AC120	Eventual
		AC121	Software fix
		AC122	Update project documentation
		AC123	Operator training
		AC124	Testware fix
		AC125	Outside vendor/Third party
	Mandatory	AC130	Deferred
		AC131	Fix in later release
		AC132	Waiver requested (reference)
	Mandatory	AC140	No fix
		AC141	No problem found
		AC142	Waiver requested (reference)
		AC143	Fix not justifiable
		AC144	Fix not identifiable
		AC145	Obsolete

Table 5b—Action classification scheme—Corrective action

Category	Compliance required	Code	Classification
Corrective action AC200	*Optional*	AC210	Department action
		AC211	Revise process (policies/procedures)
		AC212	Implement training
		AC213	Create/Revise/Reinforce standards/ specifications
		AC214	Reallocate people/resources
		AC215	Improve/Enforce audit activities
		AC220	Corporate action
		AC221	Revise process (policies/procedures)
		AC222	Implement training
		AC223	Create/Revise/Reinforce standards/ specifications
		AC224	Reallocate people/resources
		AC225	Improve/Enforce audit activities
		AC230	Industry/Government
		AC231	Sponsor research/education programs
		AC232	Compile/Publish data
		AC233	Create/Revise/Reinforce standards/ specifications
		AC234	Improve/Enforce audit activities
		AC240	Institutions for research/education
		AC241	Research problem
		AC242	Develop new technologies
		AC243	Test alternate approaches
		AC244	Create/Revise tests
		AC245	Enforce educational standards

4.2.5.1 Action supporting data items

In addition to classifying the *Resolution* and *Corrective action*, there are additional details that may be recorded. Potential supporting data items are listed in table 6. These supporting data items are optional.

Table 6—Action supporting data items

Resolution identification	Resolution action	Corrective action
Item to be fixed	Date resolution complete	Standards/Policies/Procedures to be revised
Name	Software	Organization assigned corrective action follow-up
ID number	Documentation	Person assigned corrective action follow-up
Revision	Version/Revision level	Correction action report number
Component(s) within item	Organization assigned to verify resolution	
Name	Person assigned resolution	
ID number		
Revision		
Text describing fix		
Planned date for action items' completion		
Person assigned action items		
Name		
Code or functional area		
E-mail address		
Address		
Phone number		
Planned date of fix completion		
Deferred fix of reference document		
Name		
ID number		
Revision		

4.2.6 Impact

The impact of an anomaly shall be considered at each step of the anomaly process. Impact categories classify the effect upon the product by implementing the modification requested. The *Severity* of an anomaly may be re-evaluated throughout the recognition, investigation, action, and disposition steps of the anomaly process. Identifying the *Severity* of an anomaly is a mandatory category as is identifying the *Project schedule* and *Project cost* impacts of any possible solutions for the anomaly.

Additional categories that may be useful are *Customer value* and *Priority.* The *Customer value* describes the importance to the customer(s) or potential market value of a solution to an anomaly. The *Priority* is a subjective ranking of the anomaly. It takes into account all the other impact categories.

Other impacts that may be applicable to critical or safety system software are the *Mission/Safety*, *Project risk*, *Project quality/Reliability*, and *Societal* impact categories.

Impact Categories and Classification Codes all have an IM prefix, and are listed in tables 7a–7i.

Table 7a—Impact classification scheme—Severity

Category	Compliance required	Code	Classification
Severity IM100	*Mandatory*	IM110	Urgent
		IM120	High
		IM130	Medium
		IM140	Low
		IM150	None

Table 7b—Impact classification scheme—Priority

Category	Compliance required	Code	Classification
Priority IM200	*Optional*	IM210	Urgent
		IM220	High
		IM230	Medium
		IM240	Low
		IM250	None

Table 7c—Impact classification scheme—Customer value

Category	Compliance required	Code	Classification
Customer value IM300	*Optional*	IM310	Priceless
		IM320	High
		IM330	Medium
		IM340	Low
		IM350	None
		IM360	Detrimental

Table 7d—Impact classification scheme—Mission safety

Category	Compliance required	Code	Classification
Mission safety *IM400*	*Optional*	IM410	Urgent (Prevent completion of mission or jeopardizes personnel safety)
		IM420	High (Adversely affects completion of mission, no workaround solution exists)
		IM430	Medium (Adversely affects completion of mission, workaround solution exists)
		IM440	Low (Inconvenience or annoyance)
		IM450	None of the above

Table 7e—Impact classification scheme—Project schedule

Category	Compliance required	Code	Classification
Project schedule *IM500*	*Mandatory*	IM510	High
		IM520	Medium
		IM530	Low
		IM540	None

Table 7f—Impact classification scheme—Project cost

Category	Compliance required	Code	Classification
Project cost *IM600*	*Mandatory*	IM610	High
		IM620	Medium
		IM630	Low
		IM640	None

Table 7g—Impact classification scheme—Project risk

Category	Compliance required	Code	Classification
Project risk *IM700*	*Optional*	IM710	High
		IM720	Medium
		IM730	Low
		IM740	None

Table 7h—Impact classification scheme—Project quality/reliability

Category	Compliance required	Code	Classification
Project quality/reliability IM800	*Optional*	IM810	High
		IM820	Medium
		IM830	Low
		IM840	None

Table 7i—Impact classification scheme—Societal

Category	Compliance required	Code	Classification
Societal IM900	*Optional*	IM910	High
		IM920	Medium
		IM930	Low
		IM940	None

4.2.6.1 Impact supporting data items

Potential supporting data items for recording the impact of an anomaly are listed in table 8.

Table 8—Impact supporting data items

Project impact	
Cost	Analysis
	Fix implementation (Estimated)
	Fix not done (Estimated)
	Resolution (Actual final cost)
Time	Required for fix (Estimated)
	Required for verification (Estimated)
	Impact if fix not done (Estimated)
	Required to implement (Actual final time)
Risk	Text description of risk
Schedule	If resolved
	If not resolved
	Resolution (Actual final schedule)
Contract change	Class I or Class II

4.2.7 Disposition

The final process shown in figure 1 is the disposition step. Throughout the life of an anomaly the *Disposition* category is left blank, which implies that the anomaly is open or active. When an anomaly is disposed of, it shall either be closed or designated as one of three other dispositions—deferred to a later date or release, merged with another anomaly, and referred to another project. Disposition categories and classification codes all have a DP prefix, and are listed in table 9.

Table 9—Disposition classification scheme

Category	Compliance required	Code	Classification
Disposition *DP100*	*Mandatory*	DP110	Closed
		DP111	Resolution implemented
		DP112	Not a problem
		DP113	Not in scope of project (unresolvable)
		DP114	Outside vendor's problem (reference)
		DP115	Duplicate problem (reference)
		DP120	Deferred (reference)
		DP130	Merged with another problem (reference)
		DP140	Referred to another project (reference)

4.2.7.1 Disposition supporting data items

Supporting data items that may be useful for recording the disposition of an anomaly are listed in table 10.

Table 10—Disposition supporting data items

Anomaly disposition	Verification
Action implemented	Name
Date report closed	Date
Date document update complete	Version/Revision level
Customer notified	Method
Person sending notice	Test case
Date	
Type of notice	
Reference document number (vendor, duplicate, deferred, merged, or waiver anomalies)	

Annex A
Example anomaly report

(informative)

This annex describes an example for implementing this standard within an anomaly reporting database system. The ability to rapidly retrieve data and construct reports using some or all aspects of the classification scheme is a powerful quality analysis tool. The example presented assumes that the anomaly reporting system is built upon a relational database management system (RDBMS) that provides the basic functions of querying, paging through each hit from a query (backward and forward, previous and next), adding new records, updating records, deleting records, moving to specified database tables, printing information, and developing reports. Keeping anomaly report data in a relational database facilitates the generation of reports, statistics, and information for trend analysis. Note that there are several commercial products available for anomaly tracking that can be customized to conform to this standard, in addition to the possibility of modifying an existing anomaly tracking database to conform to this standard.

The main RDBMS table contains classifications and supporting data items that generally will not be modified during the anomaly process. Figure A.1 shows the first of two screens for the anomaly input form. The square brackets mark each field in the main table.The *AR Number* (Anomaly Report Number) through *Actual Time* test fields are single entry items. The fields can be updated, but a history of changes is not maintained.

```
Query Next Previous Add Update Remove Table Screen Current Master Detail
Output Exit
_____________________ANOMALY REPORT FORM_____________________page 1 of 2__

AR Number:   [     ]                    Date Entered: [          ]
                                        Originator:   [                   ]
Project:     [            ]             Computer:     [            ]
Program:     [                   ]      Command:      [                   ]

Title:         [                                                        ]
Activity:      [   ]    Phase:             [   ]     Symptom:        [   ]
Repeatability: [   ]    Product Condition: [   ]     Customer Value: [   ]

Actual Cause: [   ]     Source:          [   ]       Type:      [   ]
Resolution:   [   ]     Schedule Impact: [   ]       Severity:  [   ]
                                                     Priority:  [   ]
Estimated Time -  Prog: [     ]   Doc: [     ]   Test: [     ]
Actual Time(hrs)- Prog: [     ]   Doc: [     ]   Test: [     ]

                        PROCESS / DISPOSITION HISTORY
Process:                          Date Process Changed:
Product Version:                  Responsible Person:
```

Figure A.1—Example anomaly report—Screen 1, main table

The classification data entered during the recognition process step are *Activity, Phase, Symptom, Repeatability,* and *Customer Value* (*Customer Value* is an impact category). In the form there is room to enter the three-digit code from the standard (minus the two-letter process step identifier). Ideally, the anomaly report system provides the user with the list of the possible classifications and codes for each category.

The *Actual Cause, Source, Type, Resolution, Schedule Impact, Severity,* and *Priority* are all categories classified during the investigation through resolution process steps. The estimated time for programming, docu-

menting and testing supporting data items is entered during the investigation process. The actual time for programming, documenting, and testing supporting data items is entered and possibly updated during the action and disposition processes.

The process/disposition history section contains data for maintaining a change history. The purpose of the history RDBMS table is to track the anomalies through the anomaly process steps, and then to identify the disposition chosen for the anomaly. Each time the process step changes, the date of the change, the version number of the affected product, and the person or group responsible for the next action are also recorded. These steps should reflect the anomaly processing mechanism in use for the project. At any time the anomaly's history can be reviewed.

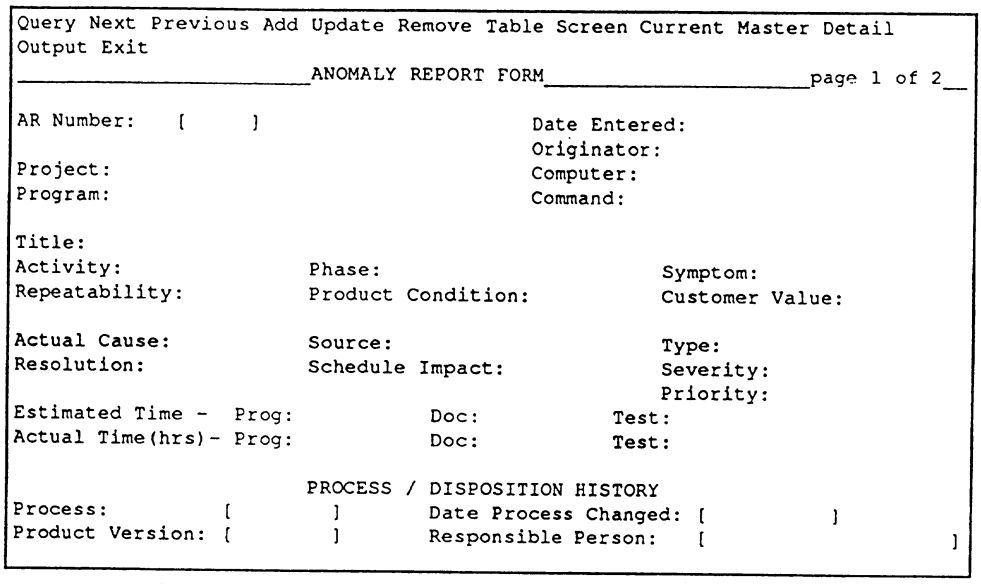

Figure A.2—Example anomaly report—Screen 1, history table

The second screen of the anomaly reporting system (see figure A.3) may also have multiple records relating to the main RDBMS table entry. This screen is used to enter and display comments and other supporting data items for the anomaly. There will probably be comments associated with each step of the anomaly process to provide details on what the anomaly is, how it happened, in what environment it occurred, what should be done about it, what was done about it, suggested corrective actions, and why the particular change involved was implemented. Multiple records in this table associated with an anomaly ensure that all information about an anomaly is maintained.

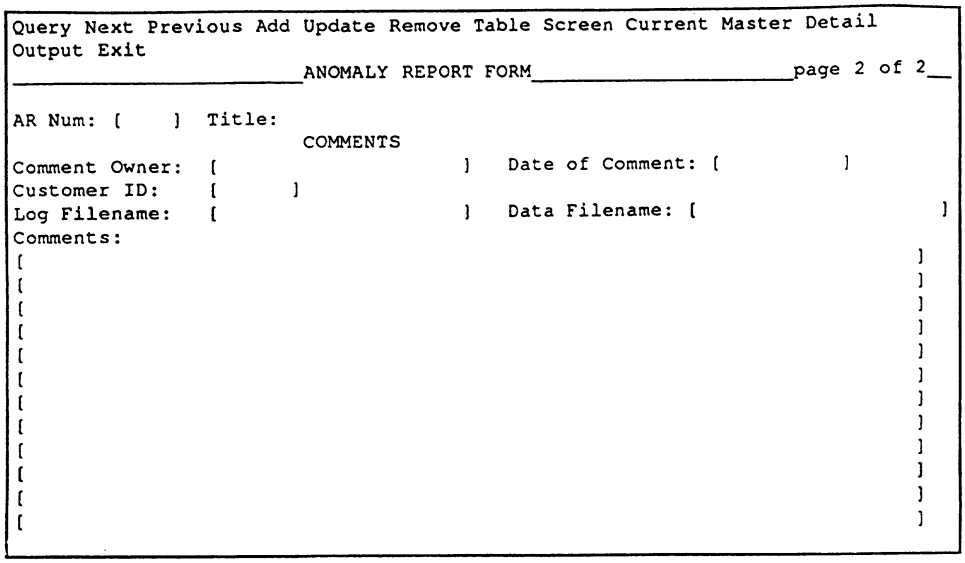

Figure A.3—Example anomaly report—Screen 2, comment table

The *AR Number* and *Title* are displayed from the first screen to provide continuity while entering data. The *Customer ID* field provides a mechanism for tracking anomalies reported by specific customers.