

IEEE Std 1219-1998

(Revision of
IEEE Std 1219-1992)

IEEE Std 1219-1998

IEEE Standard for Software Maintenance

IEEE Computer Society

Sponsored by the
Software Engineering Standards Committee

21 October 1998

SH94568

IEEE Std 1219-1998

(Revision of
IEEE Std 1219-1992)

IEEE Standard for Software Maintenance

Sponsor

**Software Engineering Standards Committee
of the
IEEE Computer Society**

Approved 25 June 1998

IEEE-SA Standards Board

Abstract: The process for managing and executing software maintenance activities is described.

Keywords: life cycle, maintenance, software, software maintenance

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

Copyright © 1998 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1998. Printed in the United States of America.

ISBN 0-7381-0336-5

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

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

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

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

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

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

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

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

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

Introduction

(This introduction is not a part of IEEE Std 1219-1998, IEEE Standard for Software Maintenance.)

This standard describes the process for managing and executing software maintenance activities. Clause 2 lists references to other standards useful in applying this standard. Clause 3 provides a set of definitions and acronyms that are either not found in other standards, or have been modified for use with this standard. Clause 4 contains required information pertaining to the software maintenance process. In order to be in compliance with this standard, Clause 4 must be adhered to.

Annex A, Maintenance guidelines, contains additional information that is not required for compliance. Topics in this annex include: the source of maintenance forms discussed in the standard, validation and verification (V&V), software quality assurance, risk assessment, safety, security, software configuration management (SCM), metrics, software replacement policy, and the maintenance process. Annex B, Supporting maintenance technology, includes the topics of re-engineering, reverse engineering, reuse, maintenance planning, impact analysis, and software tools.

The readers of this standard are referred to Annexes C and D for guidelines for using this standard to meet the requirements of IEEE/EIA 12207.1-1997, IEEE/EIA Guide for Information Technology—Software life cycle processes—Life cycle data.

The audience for which this standard is intended consists of software development managers, maintainers, software quality assurance personnel, SCM personnel, programmers, and researchers.

Participants

This standard was prepared by the Life Cycle Data Harmonization Working Group of the Software Engineering Standards Committee of the IEEE Computer Society. At the time this standard was approved, the working group consisted of the following members:

Leonard L. Tripp, *Chair*

Edward Byrne
Paul R. Croll
Perry DeWeese
Robin Fralick
Marilyn Ginsberg-Finner
John Harauz
Mark Henley

Dennis Lawrence
David Maibor
Ray Milovanovic
James Moore
Timothy Niesen
Dennis Rilling

Terry Rout
Richard Schmidt
Norman F. Schneidewind
David Schultz
Basil Sherlund
Peter Voldner
Ronald Wade

The following persons were on the balloting committee:

Syed Ali
Theodore K. Atchinson
Leo Beltracchi
H. Ronald Berlack
Richard E. Biehl
Juris Borzovs
Audrey C. Brewer
M. Scott Buck
James E. Cardow
Enrico A. Carrara
Keith Chan
Antonio M. Cicu
Theo Clarke
Rosemary Coleman
Virgil Lee Cooper

W. W. Geoff Cozens
Paul R. Croll
Patricia W. Daggett
Gregory T. Daich
Geoffrey Darnton
Taz Daughtrey
Raymond Day
Bostjan K. Derganc
Perry R. DeWeese
James Do
Evelyn S. Dow
Carl Einar Dragstedt
Charles Droz
Sherman Eagles
Leo Egan

Richard L. Evans
William Eventoff
Richard E. Fairley
John W. Fendrich
Jay Forster
Roger U. Fujii
Adel N. Ghannam
Marilyn Ginsberg-Finner
John Garth Glynn
Julio Gonzalez-Sanz
L. M. Gunther
David A. Gustafson
Jon D. Hagar
John Harauz
Robert T. Harley

Herbert Hecht
 Manfred Hein
 Debra Herrmann
 Umesh P. Hiriyannaiah
 John W. Horch
 George Jackelen
 Frank V. Jorgensen
 Vladan V. Jovanovic
 William S. Junk
 Chris F. Kemerer
 Ron S. Kenett
 Judith S. Kerner
 Robert J. Kierzyk
 Shaye Koenig
 Thomas M. Kurihara
 John B. Lane
 J. Dennis Lawrence
 Stanley H. Levinson
 Fang Ching Lim
 William M. Lively
 John Lord
 Stan Magee
 David Maibor
 Harold Mains
 Robert A. Martin

Patrick D. McCray
 Sue McGrath
 Jerome W. Mersky
 Bret Michael
 Alan Miller
 Celia H. Modell
 James W. Moore
 Pavol Navrat
 Myrna L. Olson
 Indradeb P. Pal
 Lalit M. Patnaik
 David E. Peercy
 Alex Polack
 Peter T. Poon
 Lawrence S. Przybylski
 Kenneth R. Ptack
 Ann E. Reedy
 Terence P. Rout
 Andrew P. Sage
 Stephen R. Schach
 Hans Schaefer
 Norman Schneidewind
 David J. Schultz
 Robert W. Shillato
 David M. Siefert
 Lynn J. Simms

Carl A. Singer
 Nancy M. Smith
 Alfred R. Sorkowitz
 Donald W. Sovo
 Luca Spotorno
 Fred J. Strauss
 Toru Takeshita
 Richard H. Thayer
 Booker Thomas
 Patricia Trellue
 Theodore J. Urbanowicz
 Glenn D. Venables
 Andre Villas-Boas
 Udo Voges
 Dolores Wallace
 William M. Walsh
 John W. Walz
 Camille S. White-Partain
 Scott A. Whitmire
 P. A. Wolfgang
 Paul R. Work
 Natalie C. Yopconka
 Janusz Zalewski
 Geraldine Zimmerman
 Peter F. Zoll

When the IEEE-SA Standards Board approved this standard on 25 June 1998, it had the following membership:

Richard J. Holleman, *Chair*

Donald N. Heirman, *Vice Chair*

Judith Gorman, *Secretary*

Satish K. Aggarwal
 Clyde R. Camp
 James T. Carlo
 Gary R. Engmann
 Harold E. Epstein
 Jay Forster*
 Thomas F. Garrity
 Ruben D. Garzon

James H. Gurney
 Jim D. Isaak
 Lowell G. Johnson
 Robert Kennelly
 E. G. "Al" Kiener
 Joseph L. Koepfinger*
 Stephen R. Lambert
 Jim Logothetis
 Donald C. Loughry

L. Bruce McClung
 Louis-François Pau
 Ronald C. Petersen
 Gerald H. Peterson
 John B. Posey
 Gary S. Robinson
 Hans E. Weinrich
 Donald W. Zipse

*Member Emeritus

Valerie E. Zelenty
IEEE Standards Project Editor

Contents

1. Overview.....	1
1.1 Scope.....	1
1.2 Terminology.....	1
1.3 Conventions	2
2. References.....	2
3. Definitions and acronyms	3
3.1 Definitions.....	3
3.2 Acronyms	4
4. Software maintenance	5
4.1 Problem/modification identification, classification, and prioritization	5
4.2 Analysis.....	8
4.3 Design	11
4.4 Implementation	12
4.5 System test	14
4.6 Acceptance test	15
4.7 Delivery.....	17
Annex A (informative) Maintenance guidelines	18
Annex B (informative) Supporting maintenance technology	34
Annex C (normative) Maintenance plan guidelines.....	39
Annex D (informative) Guidelines for compliance with IEEE/EIA 12207.1-1997	43

IEEE Standard for Software Maintenance

1. Overview

1.1 Scope

This standard describes an iterative process for managing and executing software maintenance activities. Use of this standard is not restricted by size, complexity, criticality, or application of the software product. This standard uses the process model, depicted in Table 2, to discuss and depict each phase of software maintenance. The criteria established apply to both the planning of maintenance for software while under development, as well as the planning and execution of software maintenance activities for existing software products. Ideally, maintenance planning should begin during the stage of planning for software development (see A.3 for guidance).

This standard prescribes requirements for process, control, and management of the planning, execution, and documentation of software maintenance activities. In totality, the requirements constitute a minimal set of criteria that are necessary and sufficient conditions for conformance to this standard. Users of this standard may incorporate other items by reference or as text to meet their specific needs.

The basic process model includes input, process, output, and control for software maintenance. Metrics/measures captured for maintenance should enable the manager to manage the process and the implementor to implement the process (see Table 3). This standard does not presuppose the use of any particular development model (e.g., waterfall, spiral, etc.).

This standard provides additional software maintenance guidance on associated topics in Annex A and tools/technology assistance in Annex B. Maintenance plan guidelines are provided in Annex C and guidelines for compliance with IEEE/EIA 12207.1-1997 are provided in Annex D.

1.2 Terminology

The words *shall* and *must* identify the mandatory (essential) material within this standard. The words *should* and *may* identify optional (conditional) material. The terminology in this standard is based on IEEE Std 610.12-1990.¹ New terms and modified definitions as applied in this standard are included in Clause 3.

¹ Information on references can be found in Clause 2.

1.3 Conventions

The conventions used in each figure depicting a maintenance phase are shown in Figure 1.

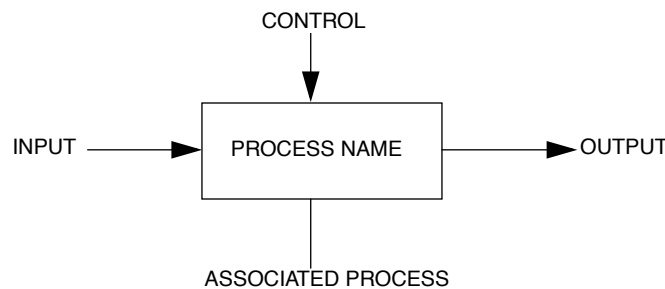


Figure 1—Conventions

The term *associated processes* refers to external processes that are defined in other standards; i.e., software quality assurance (SQA), software configuration management (SCM), and verification and validation (V&V). The term *associated processes* also refers to the metrics process illustrated within this standard.

2. References

Table 1 provides a cross-reference of IEEE standards that address various topics related to software maintenance. This standard shall be used in conjunction with the following publications. When the following standards are superseded by an approved revision, the revision shall apply.

IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.²

IEEE Std 730-1998, IEEE Standard for Software Quality Assurance Plans.

IEEE Std 730.1-1995, IEEE Guide for Software Quality Assurance Planning.

IEEE Std 828-1998, IEEE Standard for Software Configuration Management Plans.³

IEEE Std 829-1998, IEEE Standard for Software Test Documentation.⁴

IEEE Std 982.1-1988, IEEE Standard Dictionary of Measures to Produce Reliable Software.

IEEE Std 982.2-1988, IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software.

IEEE Std 1012-1998, IEEE Standard for Software Verification and Validation.

IEEE Std 1012a-1998, IEEE Standard for Software Verification and Validation: Content Map to IEEE/EIA 12207.1-1997.⁵

IEEE Std 1028-1997, IEEE Standard for Software Reviews.

IEEE Std 1042-1987 (Reaff 1993), IEEE Guide to Software Configuration Management.

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

³As this standard goes to press, IEEE Std 828-1998, IEEE Std 829-1998, and IEEE Std 1012a-1998 are approved but not yet published. The draft standards are, however, available from the IEEE. Anticipated publication date is Fall 1998. Contact the IEEE Standards Department at 1 (732) 562-3800 for status information.

⁴See Footnote 3.

⁵See Footnote 3.

IEEE P1058/D2.1, Draft Standard for Software Project Management Plans, dated 5 August 1998.⁶

IEEE Std 1058a-1998, IEEE Standard for Software Project Management Plans: Content Map to IEEE/EIA 12207.1-1997.⁷

IEEE Std 1074-1997, IEEE Standard for Developing Software Life Cycle Processes.

Table 1—The relationship of IEEE software engineering standards to IEEE Std 1219-1998

Relationship		IEEE standard
Process	Problem ID/classification	—
	Analysis	830-1998, 1074-1997
	Design	830-1998, 1016-1998, 1074-1997
	Implementation	1008-1987, 1074-1997
	System test	829-1998, 1012-1998, 1012a-1998, 1028-1988, 1074-1997
	Acceptance testing	1012-1998, 1012a-1998, 1074-1997
	Delivery	—
Control	Problem ID/classification	—
	Analysis	830-1998
	Design	830-1998, 1016-1998
	Implementation	829-1998, 1008-1987
	System test	829-1998, 1012-1998, 1012a-1998, 1028-1988
	Acceptance testing	829-1998, 1012-1998, 1012a-1998, 1028-1988
	Delivery	1063-1987
Management	Configuration management	828-1998, 1042-1987
	Measurement/metrics	982.1-1988, 982.2-1988
	Planning	829-1998, 1012-1998, 1012a-1998, P1058/D2.1, 1058a-1998
	Tools/techniques	—
	Quality Assurance	730-1998, 730.1-1995
	Risk assessment	730-1998, 982.2-1988
	Safety	—
	Security	—

3. Definitions and acronyms

3.1 Definitions

The definitions listed below establish meaning in the context of this standard. These are contextual definitions serving to augment the understanding of software maintenance activities as described within this standard. Other definitions can be found in IEEE Std 610.12-1990.

3.1.1 adaptive maintenance: Modification of a software product performed after delivery to keep a computer program usable in a changed or changing environment.

⁶Upon approval of IEEE P1058 by the IEEE-SA Standards Board, this standard will be integrated with IEEE Std 1058a-1998 and published as IEEE Std 1058, 1998 Edition. Approval is expected 8 December 1998.

⁷See Footnote 6.

3.1.2 corrective maintenance: Reactive modification of a software product performed after delivery to correct discovered faults.

3.1.3 customer: The person, or persons, for whom the product is intended, and usually (but not necessarily) who decides the requirements.

3.1.4 emergency maintenance: Unscheduled corrective maintenance performed to keep a system operational.

3.1.5 interoperability testing: Testing conducted to ensure that a modified system retains the capability of exchanging information with systems of different types, and of using that information.

3.1.6 modification request (MR): A generic term that includes the forms associated with the various trouble/problem-reporting documents (e.g., incident report, trouble report) and the configuration change control documents [e.g., software change request (SCR)].

3.1.7 perfective maintenance: Modification of a software product after delivery to improve performance or maintainability.

3.1.8 project: A subsystem that is subject to maintenance activity.

3.1.9 regression test: Retesting to detect faults introduced by modification.

3.1.10 repository: (A) A collection of all software-related artifacts (e.g., the software engineering environment) belonging to a system. (B) The location/format in which such a collection is stored.

3.1.11 reverse engineering: The process of extracting software system information (including documentation) from source code.

3.1.12 software maintenance: Modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment.

3.1.13 system: A set of interlinked units organized to accomplish one or several specific functions.

3.1.14 user: The person or persons operating or interacting directly with the system.

3.2 Acronyms

The following acronyms are referred to in this standard:

CARE	computer-assisted re-engineering
CASE	computer-aided software engineering
CM	configuration management
CPU	central processing unit
CSA	configuration status accounting
FCA	functional configuration audit
FR	feasibility report
I/O	input/output
LC	linear circuit
MP	maintenance plan
MR	modification request
PCA	physical configuration audit
PDL	program design language
SCA	software change authorization

SCM	software configuration management
SCR	system/software change request
SE	software engineering
SLOC	source lines of code
SQA	software quality assurance
V&V	verification and validation
VDD	version description document

4. Software maintenance

This standard defines changes to the software process through a defined maintenance process that includes the following phases:

- a) Problem/modification identification, classification, and prioritization;
- b) Analysis;
- c) Design;
- d) Implementation;
- e) Regression/system testing;
- f) Acceptance testing; and
- g) Delivery.

These phases are graphically depicted in Table 2. Software maintenance factors in Table 3 are the entities qualified by the associated metrics/measures identified for each phase.

4.1 Problem/modification identification, classification, and prioritization

In this phase, software modifications are identified, classified, and assigned an initial priority ranking. Each modification request (MR) shall be evaluated to determine its classification and handling priority. Classification shall be identified from the following maintenance types:

- a) Corrective;
- b) Adaptive;
- c) Perfective; and
- d) Emergency.

Metrics/measures and associated factors identified for this phase should be collected and reviewed at appropriate intervals (see Table 3 and IEEE Std 982.1-1988 and IEEE Std 982.2-1988).

4.1.1 Input

Input for the problem/modification identification and classification phase shall be an MR.

4.1.2 Process

If a modification to the software is required, the following determinative activities shall occur within the maintenance process:

- a) Assign an identification number;
- b) Classify the type of maintenance;
- c) Analyze the modification to determine whether to accept, reject, or further evaluate;
- d) Make a preliminary estimate of the modification size/magnitude;
- e) Prioritize the modification; and
- f) Assign an MR to a block of modifications scheduled for implementation.

Table 2—Process model for software maintenance

	Problem identification	Analysis	Design	Implementation	System test	Acceptance test	Delivery
Input	MR	Project/system document Repository information Validated MR	Project/system document Source code Databases Analysis phase output	Source code Product/system document Results of design phase	Updated software documentation Test-readiness review report Updated system	Test-readiness review report Fully integrated system Acceptance test • Plans • Cases • Procedures	Tested/accepted system
Process	Assign change number Classify Accept or reject change Preliminary magnitude estimate Prioritize	Feasibility analysis Detailed analysis Redocument, if needed	Create test cases Revise • Requirements • Implementation plan	Code Unit test Test-readiness review	Functional test Interface testing Regression testing Test-readiness review	Acceptance test Interoperability test	PCA Install Training
Control	Uniquely identify MR Enter MR into repository	Conduct technical review Verify • Test strategy • Documentation is updated Identify safety and security issues	Software inspection/review Verify design	Software inspection/review Verify • CM control of software • Traceability of design	CM control of • Code • Listings • MR • Test documentation	Acceptance test Functional audit Establish baseline	PCA VDD
Output	Validated MR Process determinations	Feasibility report (FR) Detailed analysis report Updated requirements Preliminary modification list Implementation plan Test strategy	Revised • Modification list • Detail analysis • Implementation plan Updated • Design baseline • Test plans	Updated • Software • Design documents • Test documents • User documents • Training material Test-readiness review report	Tested system Test reports	New system baseline Acceptance test report FCA report	PCA report VDD
Metrics	See Table 3						

Table 3— Process model metrics for software maintenance

	Problem identification	Analysis	Design	Implementation	System test	Acceptance test	Delivery
Factors	Correctness Maintainability	Flexibility Traceability Reusability Usability Maintainability Comprehensibility	Flexibility Traceability Reusability Testability Maintainability Comprehensibility Reliability	Flexibility Traceability Maintainability Comprehensibility Reliability	Flexibility Traceability Verifiability Testability Interoperability Comprehensibility Reliability	Flexibility Traceability Interoperability Testability Comprehensibility Reliability	Completeness Reliability
Metrics	No. of omissions on MR No. of MR submittals No. of duplicate MRs Time expended for problem validation	Requirement changes Documentation error rates Effort per function area (SQA, SE, etc.) Elapsed time (schedule) Error rates generated by priority and type	Software complexity Design changes Effort per function area Elapsed time Test plans and procedure changes Error rates generated by priority and type Number of lines of code added, deleted, modified, tested Number of applications	Volume/functionality (function points or source lines of code) Error rates generated by priority and type	Error rates by priority and type • Generated • Corrected	Error rates by priority and type • Generated • Corrected	Documentation changes (i.e., VDDs, training manuals, operation guidelines)

Figure 2 summarizes the input, process, control, and output for the problem/modification identification and classification phase of maintenance. For additional information, see also A.4.1.

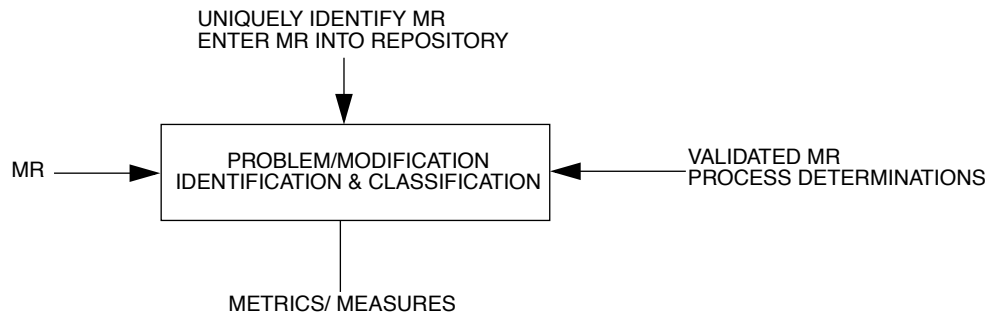


Figure 2— Problem/modification identification and classification phase

4.1.3 Control

MR and process determinations shall be uniquely identified and entered into a repository. See also A.11.1 for guidance.

4.1.4 Output

The output of this process shall be the validated MR and the process determinations that were stored in a repository. The repository shall contain the following items:

- a) Statement of the problem or new requirement;
- b) Problem or requirement evaluation;
- c) Classification of the type of maintenance required;
- d) Initial priority;
- e) Verification data;
- f) Initial estimate of resources required to modify the existing system.

4.2 Analysis

The analysis phase shall use the repository information and the MR validated in the modification identification and classification phase, along with system and project documentation, to study the feasibility and scope of the modification and to devise a preliminary plan for design, implementation, test, and delivery. Metrics/ measures and associated factors identified for this phase should be collected and reviewed at appropriate intervals (see Table 3 and IEEE Std 982.1-1988 and IEEE Std 982.2-1988).

Figure 3 summarizes the input, process, control, and output for the analysis phase of maintenance. For additional guidance, see A.4.2.

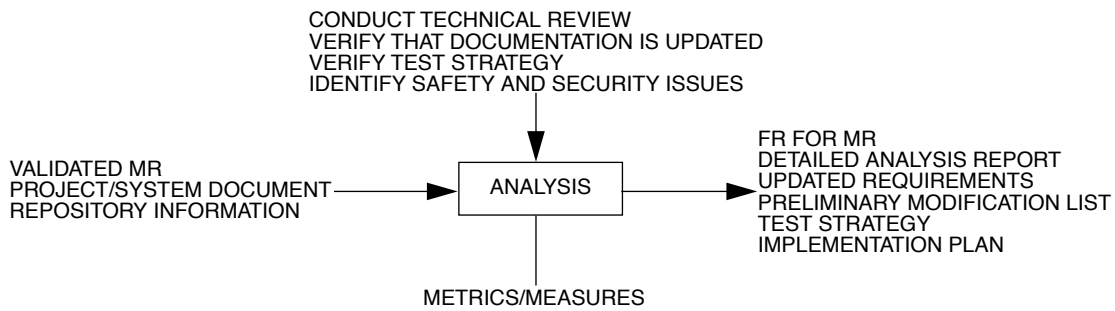


Figure 3—Analysis phase

4.2.1 Input

The input to the analysis phase of the maintenance process shall include the following:

- a) Validated MR;
- b) Initial resource estimate and other repository information;
- c) Project and system documentation, if available.

4.2.2 Process

Analysis is an iterative process having at least two components:

- a) A feasibility analysis; and
- b) A detailed analysis.

If the documentation is not available or is insufficient and the source code is the only reliable representation of the software system, reverse engineering is recommended (see Annex B for guidance).

4.2.2.1 Feasibility analysis

A feasibility analysis shall be performed for an MR and an FR shall be prepared. This FR should contain the following:

- a) Impact of the modification;
- b) Alternate solutions, including prototyping;
- c) Analysis of conversion requirements;
- d) Safety and security implications;
- e) Human factors;
- f) Short-term and long-term costs;
- g) Value of the benefit of making the modification.

4.2.2.2 Detailed analysis

Detailed analysis shall include the following:

- a) Define firm requirements for the modification;
- b) Identify the elements of modification;
- c) Identify safety and security issues (see also A.9 and A.10 for guidance);
- d) Devise a test strategy;
- e) Develop an implementation plan.

In identifying the elements of modification (creating the preliminary modification list), analysts examine all products (e.g., software, specifications, databases, documentation) that are affected. Each of these products shall be identified, and generated if necessary, specifying the portions of the product to be modified, the interfaces affected, the user-noticeable changes expected, the relative degree and kind of experience required to make changes, and the estimated time to complete the modification.

The test strategy is based on input from the previous activity identifying the elements of modification. Requirements for at least three levels of test, including individual element tests, integration tests, and user-oriented functional acceptance tests shall be defined. Regression test requirements associated with each of these levels of test shall be identified as well. The test cases to be used for testing to establish the test base-line shall be revalidated.

A preliminary implementation plan shall state how the design, implementation, testing, and delivery of the modification is to be accomplished with a minimal impact to current users.

4.2.3 Control

Control of analysis shall include the following:

- a) Retrieval of the relevant version of project and system documentation from the configuration control function of the organization;
- b) Review of the proposed changes and engineering analysis to assess technical and economic feasibility, and assess correctness;
- c) Identification of safety and security issues;
- d) Consideration of the integration of the proposed change within the existing software;
- e) Verification that all appropriate analysis and project documentation is updated and properly controlled;
- f) Verification that the test function of the organization is providing a strategy for testing the change(s), and that the change schedule can support the proposed test strategy;
- g) Review of the resource estimates and schedules and verification of their accuracy;
- h) Technical review to select the problem reports and proposed enhancements to be implemented in the new release. The list of changes shall be documented.

Consult A.6, A.7, and A.11.2 for guidance on activities related to V&V, SQA, and SCM.

At the end of the analysis phase, a risk analysis shall be performed (see also A.8 for guidance). Using the output of the analysis phase, the preliminary resource estimate shall be revised, and a decision, that includes the customer, is made on whether to proceed to the design phase.

4.2.4 Output

The output of the maintenance process analysis phase shall include the following:

- a) FR for MRs;
- b) Detailed analysis report;
- c) Updated requirements (including traceability list);
- d) Preliminary modification list;
- e) Test strategy;
- f) Implementation plan.