# INTERNATIONAL STANDARD

# ISO/IEC 14764

## IEEE Std 14764-2006

Second edition
2006-09-01

# Software Engineering — Software Life Cycle Processes — Maintenance

*Ingénierie du logiciel — Processus du cycle de vie du logiciel — Maintenance*

Reference number
ISO/IEC 14764:2006(E)
IEEE
Std 14764-2006

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

**PDF disclaimer**

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.



ISO
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org



Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

**ISO/IEC 14764:2006(E)**
**IEEE Std 14764-2006**

(Revision of IEEE Std 1219-1988)

Secretariat: Standards Council of Canada
Date 2005-06-29

# Standard for Software Engineering — Software Life Cycle Processes — Maintenance

*Norme pour ingénierie du logiciel — Processus de cycle de vie du logiciel — Maintenance*

Sponsor
**Software & Systems Engineering Standards Committee**
of the
**IEEE Computer Society**

Approved 30 March 2006
**IEEE SA Standards Board**

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

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

**Keywords:** life cycle, maintenance, software, software maintenance



This ISO/IEEE document is an International Standard and is copyright-protected by ISO and the IEEE. Except as permitted under the applicable laws of the user's country, neither this ISO/IEEE standard nor any extract from it may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, photocopying, recording or otherwise, without prior written permission being secured. Requests for permission to reproduce should be addressed to either ISO or the IEEE at the addresses below.

ISO Copyright Office
Case postale 56 · CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Institute of Electrical and Electronics Engineers
Standards Association
Manager, Standards
Intellectual Property
445 Hoes Lane
Piscataway, NJ 08854
E-mail: stds.ipr@ieee.org
Web: www.ieee.org

Copyright © 2006 ISO/IEEE. All rights reserved.
Published 19 May 2006. Printed in the United States of America.
IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Incorporated.

*Print:* ISBN 0-7381-4960-8 SH95534
*PDF:* ISBN 0-7381-4961-6 SS95534

*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.*



Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
© IEEE 2006 – All rights reserved
Not for Resale

**IEEE Standards** documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

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

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

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. 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.

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

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

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

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
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; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

--`,,```,,,,````-`-`,,`,,`,`,,`---

© IEEE 2006 All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

# International Standard ISO/IEC 14764:2006(E)

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC/IEEE 14764 was prepared by Joint Technical Committee ISO/IEC JTC 1, Subcommittee SC 7, (Software and System Engineering).

The first edition of ISO/IEC 14764 was prepared by ISO/IEC JTC 1/SC 7. The current edition is the result of merging the original edition with IEEE Std 1219-1998. ISO/IEC JTC 1/SC 7 and the IEEE Computer Society cooperated in this project to merge the two standards. This second edition cancels and replaces the first edition (1999).



International Organization for Standardization/International Electrotechnical Commission
Case postale 56 • CH-1211 Genève 20 • Switzerland



Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

© IEEE 2006 – All rights reserved

# IEEE Introduction

This introduction is not part of ISO/IEC/IEEE 14764:2005(E), Standard for Software Engineering—Software Life Cycle Processes—Maintenance.

This International Standard provides guidance on the Software Maintenance Process. Software Maintenance is a primary process in the life cycle of a software product, as described in ISO/IEC 12207, "Information technology – Software life cycle processes." The Maintenance Process contains the activities and tasks of the maintainer. This International Standard is part of the ISO/IEC 12207 family of documents. In this International Standard, ISO/IEC 12207 refers to ISO/IEC 12207:1995 as amended in 2002 and 2004. The only mandatory clauses in this International Standard come from ISO/IEC 12207. The mandatory clauses contain shalls and each shall from ISO/IEC 12207 that is duplicated in this International Standard is boxed. The related ISO/IEC 12207 clause number is listed after the boxed ISO/IEC 12207 shalls. This International Standard is the result of the harmonization of ISO/IEC 14764 and IEEE Std 1219-1998.[1]

Because maintenance consumes a major share of a software life cycle financial resources, it should be an important project consideration.

During operation of the software, problems may be detected that were not detected during validation and acceptance. Therefore, a maintenance effort is needed to cope with these problems. This maintenance effort also covers software improvements needed to meet new or modified user requirements. Software maintenance is commonly needed when upgrading system components, such as operating systems and databases, as well as when modifications are made to external software and systems interfaces. Software maintenance may be a significant portion of life cycle costs.

Software maintainers use a number of specific tools, methods, and techniques. This International Standard does not specify how to implement or perform the activities and tasks in the Software Maintenance Process since these are dependent upon the formal agreement and organizational requirements. Maintenance is required on all types of software, whatever the technology, technique, or tool used to create it.

Clause 1 provides the scope of this International Standard. Clause 2 provides conformance information. Clause 3 provides normative references. Clause 4 provides terms and definitions. Clause 5 provides the application of this International Standard. Clause 6 provides the details of the Maintenance Process. Clause 7 provides execution considerations for the Maintenance Process. Clause 8 provides the software maintenance strategy. Annex A provides a cross reference between clauses in this International Standard and ISO/IEC 12207. Annex B provides a list of abbreviations used in this International Standard. Annex C provides a bibliography.

## Notice to users

### Errata

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

### Interpretations

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

### Patents

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

[1] IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

© IEEE 2006 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

## Participants

At the time this standard was completed, the Software Maintenance Working Group had the following membership:

**Paul Croll,** *Chair*
**Thomas Pigoski,** *Editor*
**James W. Moore,** *Liaison to ISO/IEC JTC 1/SC 7*

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

Edward Bartlett
Bakul Banerjee
Juris Borzovs
Curtis Browne
Bruce Bullock
Joseph Butchko
Keith Chow
Antonio M. Cicu
Todd Cooper
Paul Croll
Geoffrey Darnton
Guru Dutt Dhingra
Einar Dragstedt
Clint Early
Christof Ebert
John Fendrich
Gregg Giesler
John Garth Glynn
Lewis Gray
Michael Grimley
Victoria Hailey
John Horch
Peeya Iwagoshi
William Junk
Joerg Kampmann
Piotr Karocki
Ron Kenett
J. Dennis Lawrence
Carol Long
James Moore
Jeremy Moore
Richard Moore
Rajesh Moorkath
Gerald Radack
Annette Reilly
Garry Roedler
Terence Rout
James Ruggieri
James Sanders
Robert J. Schaaf
David Schultz
Carl Singer
Mike Smith
Mitchell Smith
Luca Spotorno
Thomas Starai
Richard Thayer
Scott Valcourt
John Walz
Oren Yuen
Janusz Zalewski
Li Zhang
Geraldine Zimmerman

When the IEEE-SA Standards Board approved this standard on 30 March 2006, it had the following membership:

**Steve M. Mills,** *Chair*
**Richard H. Hulett,** *Vice Chair*
**Don Wright,** *Past Chair*
**Judith Gorman,** *Secretary*

Mark D. Bowman
Dennis B. Brophy
Joseph Bruder
Richard Cox
Bob Davis
Julian Forster*
Joanna N. Guenin
Mark S. Halpin
Raymond Hapeman
William B. Hopf
Lowell G. Johnson
Herman Koch
Joseph L. Koepfinger*
David J. Law
Daleep C. Mohla
Paul Nikolich
T. W. Olsen
Glenn Parsons
Ronald C. Petersen
Gary S. Robinson
Frank Stone
Malcolm V. Thaden
Richard L. Townsend
Joe D. Watson
Howard L. Wolfman

*Member Emeritus

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

Satish K. Aggarwal, *NRC Representative*
*Richard DeBlasio, DOE Representative*
*Alan H. Cookson, NIST Representative*

Mike Fisher
*IEEE Standards Project Editor*

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
© IEEE 2006 – All rights reserved
Not for Resale

# CONTENTS

1 Overview ..... 1
1.1 Scope ..... 1
1.2 Purpose ..... 1
1.3 Field of application ..... 1
1.4 Limitations ..... 2
1.5 Conformance ..... 2

2 Normative references ..... 2

3 Definitions and terms ..... 3

4 Application of this International Standard ..... 4
4.1 Maintenance Process ..... 4
4.2 Organization of this International Standard ..... 5

5 Maintenance Processes ..... 5
5.1 Process Implementation ..... 6
5.1.1 Inputs ..... 6
5.1.2 Tasks ..... 7
5.1.3 Controls ..... 8
5.1.4 Support ..... 8
5.1.5 Outputs ..... 8
5.2 Problem and Modification Analysis ..... 9
5.2.1 Inputs ..... 9
5.2.2 Tasks ..... 9
5.2.3 Controls ..... 12
5.2.4 Support ..... 12
5.2.5 Outputs ..... 12
5.3 Modification Implementation ..... 13
5.3.1 Inputs ..... 13
5.3.2 Tasks ..... 13
5.3.3 Controls ..... 14
5.3.4 Support ..... 14
5.3.5 Outputs ..... 15
5.4 Maintenance Review/Acceptance ..... 15
5.4.1 Inputs ..... 15
5.4.2 Tasks ..... 15
5.4.3 Controls ..... 16
5.4.4 Support ..... 16
5.4.5 Outputs ..... 17
5.5 Migration ..... 17
5.5.1 Inputs ..... 17
5.5.2 Tasks ..... 17
5.5.3 Controls ..... 20
5.5.4 Support ..... 20
5.5.5 Outputs ..... 21
5.6 Software retirement ..... 21
5.6.1 Inputs ..... 21
5.6.2 Tasks ..... 22
5.6.3 Controls ..... 24
5.6.4 Support ..... 24
5.6.5 Outputs ..... 24

6 Execution considerations ..... 24
6.1 Introduction ..... 24
6.2 Types of maintenance ..... 25
6.3 Arrangements for maintenance ..... 25

© IEEE 2006 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

6.4 Tools for maintenance ....26
6.5 Software maintenance measurement ....27
6.6 Documentation of process ....27
6.7 Early involvement in development ....27
6.8 Maintainability ....28
6.8.1 Maintainability and the development process ....28
6.8.2 Maintainability and specific activities in the development process ....29
6.9 Software transition ....30
6.10 Documentation ....31

7 Software maintenance strategy ....31
7.1 Introduction ....31
7.2 The maintenance concept ....32
7.2.1 Scope ....32
7.2.2 Defining the process ....32
7.2.3 Designation of who will provide maintenance ....32
7.2.4 Estimate of maintenance costs ....33
7.3 Maintenance planning ....33
7.3.1 Introduction ....33
7.3.2 The maintenance plan ....33
7.3.3 Maintenance plan topics ....34
7.4 Resource analysis ....37
7.4.1 Personnel resources ....38
7.4.2 Environment resources ....38
7.4.3 Financial resources ....38

Annex A (informative) Cross-reference between ISO/IEC/IEEE 14764 and ISO/IEC 12207 and ISO/IEC 12207 Amd 1 ....39

Annex B (informative) Abbreviations ....42

Annex C (informative) Bibliography ....43

--`,,```,,,,````-`-`,,`,,`,`,,`---

© IEEE 2006 – All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS

Not for Resale

# Standard for Software Engineering — Software Life Cycle Processes — Maintenance

## 1 Overview

This International Standard describes in greater detail management of the Maintenance Process described in ISO/IEC 12207, including Amendments. This International Standard also establishes definitions for the various types of maintenance. This International Standard provides guidance that applies to planning, execution and control, review and evaluation, and closure of the Maintenance Process. The scope of this International Standard includes maintenance for multiple software products with the same maintenance resources. “Maintenance” in this International Standard means software maintenance unless otherwise stated.

### 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 a process model 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.

This International Standard provides the framework within which generic and specific software maintenance plans may be executed, evaluated, and tailored to the maintenance scope and magnitude of given software products.

This International Standard provides the framework, precise terminology, and processes to allow the consistent application of technology (tools, techniques, and methods) to software maintenance.

This International Standard provides guidance for the maintenance of software. The basis for the Maintenance Process and its activities comes from the definitions of ISO/IEC 12207. It defines the activities and tasks of software maintenance, and provides maintenance planning requirements. It does not address the operation of software and the operational functions, e.g., backup, recovery, system administration, which are normally performed by those who operate the software.

This International Standard is written primarily for maintainers of software and additionally for those responsible for development and quality assurance. It may also be used by acquirers and users of systems containing software who may provide inputs to the maintenance plan.

### 1.2 Purpose

This International Standard provides guidance on the management of (or how to perform) the Maintenance Process. It identifies how the Maintenance Process can be invoked during acquisition and operation. This lnternational Standard also emphasizes the following in the Maintenance Process: the maintainability of software products; the need for maintenance service models; and the need for a maintenance strategy and plan.

### 1.3 Field of application

This International Standard is intended to provide guidance for the planning for and maintenance of software products or services, whether performed internally or externally to an organization. It is not intended to apply to the operation of the software.

Copyright © 2006 IEEE. All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

This International Standard is intended to provide guidance for two-party situations and may be equally applied where the two parties are from the same organization. This International Standard is intended to also be used by a single party as self-imposed tasks (ISO/IEC 12207).

This International Standard is not intended for software products that are "throw-away" or a "short-term" solution.

It is intended for self-imposition by developers of off-the-shelf software products to maintain such products. It is not intended for software products customized by users and products maintained as end-user applications. Maintenance is applied to computer programs, code, data, and documentation. It is intended to apply to software products created during the development of the software product. This may include such things as the test software, test databases, the Software Test Environment (STE), or the Software Engineering Environment (SEE).

This International Standard is intended for use in all maintenance efforts, regardless of the life cycle model (e.g., incremental, waterfall, evolutionary). This International Standard is not restricted by size, complexity, criticality, or application of the software product. This International Standard is intended to guide the use of results from the Maintenance Process as input to the next development in order to improve the maintainability of the software product.

### 1.4 Limitations

This International Standard describes the framework of the Software Maintenance Process but does not specify the details of how to implement or perform the activities and tasks included in the process.

In this International Standard, there are a number of lists. None of these is presumed to be exhaustive. They are intended as examples.

### 1.5 Conformance

This International Standard provides guidance for the execution of the Maintenance Process of ISO/IEC 12207. The guidance in this standard is completely consistent with ISO/IEC 12207. Conformance cannot be claimed to this standard but can be claimed to the ISO/IEC 12207 Maintenance Process and related tailoring.

## 2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC/IEEE 14764. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC/IEEE 14764 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 9126-1:2001, *Software engineering -- Product quality -- Part 1: Quality model.*[2]

ISO/IEC 12207:1995, *Information technology -- Software life cycle processes.*

ISO/IEC 12207: Amd 1:2002, *Information technology -- Software life cycle processes (AMENDMENT 1).*

---

[2] ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse (http://www.iso.ch/). ISO/IEC publications are also available in the United States from Global Engineering Documents, 15 Inverness Way East, Englewood, CO 80112, USA (http://global.ihs.com/). Electronic copies are available in the United States from the American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (http://www.ansi.org/).


Copyright © 2006 IEEE. All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

ISO/IEC 12207: Amd 2:2004, *Information technology -- Software life cycle processes (AMENDMENT 2).*

ISO/IEC 15939:2002, *Software engineering – Software measurement process.*

## 3 Definitions and terms

For the purpose of this standard, the following definitions apply. The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition, and the the terms and definitions given in ISO/IEC 12207 should be referenced for terms not defined in this clause.

### 3.1 adaptive maintenance

the modification of a software product, performed after delivery, to keep a software product usable in a changed or changing environment

NOTE—Adaptive maintenance provides enhancements necessary to accommodate changes in the environment in which a software product must operate. These changes are those that must be made to keep pace with the changing environment. For example, the operating system might be upgraded and some changes may be made to accommodate the new operating system.[3]

### 3.2 corrective maintenance

the reactive modification of a software product performed after delivery to correct discovered problems

NOTE—The modification repairs the software product to satisfy requirements.

### 3.3 emergency maintenance

an unscheduled modification performed to temporarily keep a system operational pending corrective maintenance

NOTE—Emergency maintenance is a part of corrective maintenance

### 3.4 maintainability

the capability of the software product to be modified. Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications [ISO/IEC 9126-1][4]

### 3.5 maintenance enhancement

a modification to an existing software product to satisfy a new requirement

NOTE—There are two types of software enhancements, adaptive and perfective. A maintenance enhancement is not a software correction.

### 3.6 Modification Request (MR)

a generic term used to identify proposed modifications to a software product that is being maintained

NOTE—The MR may later be classified as a correction or enhancement and identified as corrective, preventive, adaptive, or perfective maintenance. MRs are also referred to as change requests. See Figure 1.

---

[3] Notes in text, tables, and figures are given for information only, and do not contain requirements needed to implement the standard.

[4] Information on references can be found in Clause 2.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright © 2006 IEEE. All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

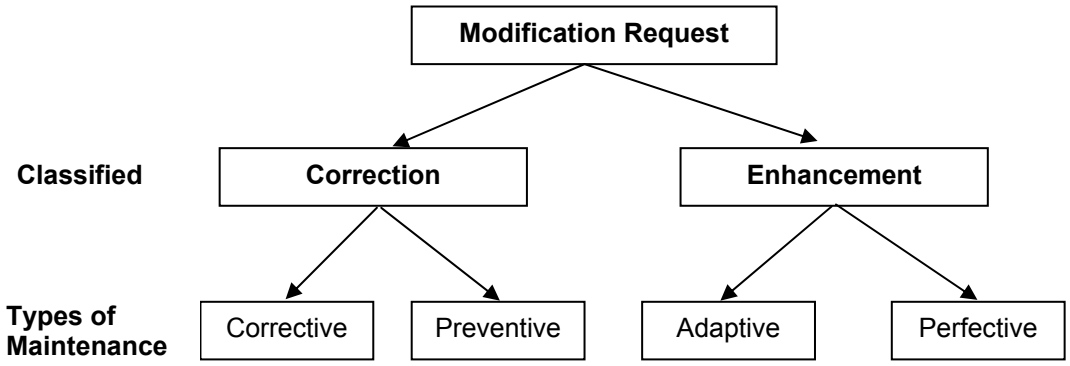



**Figure 1 — Modification Request**

### 3.7 perfective maintenance

the modification of a software product after delivery to detect and correct latent faults in the software product before they are manifested as failures

NOTE—Perfective maintenance provides enhancements for users, improvement of program documentation, and recoding to improve software performance, maintainability, or other software attributes.

### 3.8 preventive maintenance

the modification of a software product after delivery to detect and correct latent faults in the software product before they become operational faults

### 3.9 Problem Report (PR)

a term used to identify and describe problems detected in a software product

NOTE—PRs are either submitted directly to denote faults or established after impact analysis is performed on Modification Requests and faults are found.

### 3.10 software maintenance

the totality of activities required to provide cost-effective support to a software system. Activities are performed during the pre-delivery stage as well as the post-delivery stage

NOTE—Pre-delivery activities include planning for post-delivery operations, supportability, and logistics determination. Post-delivery activities include software modification, training, and operating a help desk.

### 3.11 software transition

a controlled and coordinated sequence of actions wherein software development passes from the organization performing initial software development to the organization performing software maintenance

## 4 Application of this International Standard

This clause presents the Maintenance Process that is required to maintain software products.

### 4.1 Maintenance Process

Maintenance is one of the five primary life cycle processes that may be performed during the life cycle of software (ISO/IEC 12207). The Acquisition and Supply primary life cycle processes of ISO/IEC 12207 may initiate the Process Implementation activity of the Maintenance primary life cycle process through an agreement or contract. The Operation primary life cycle process of


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

ISO/IEC 12207 may initiate the Maintenance life cycle process through submission of a Modification Request or Problem Report. The Maintenance life cycle process invokes the Development primary life cycle process of ISO/IEC 12207. The supporting processes of Documentation, Configuration Management, Quality Assurance, Verification, Validation, Joint Review, Audit, and Problem Resolution of ISO/IEC 12207 are used by the Maintenance life cycle process.

### 4.2 Organization of this International Standard

The clauses that follow are presented in the order that Maintainers should address them.

Clause 6 provides the details of the Maintenance Process including tasks and task-steps needed to implement the Maintenance Process. Clause 7 provides execution considerations, and issues to be considered when planning for maintenance. Clause 8 provides comprehensive planning information.

## 5 Maintenance Processes

This clause defines the activities and tasks for the primary life cycle process of software maintenance.

The Maintenance Process contains the activities and tasks necessary to modify an existing software product while preserving its integrity. These activities and tasks are the responsibility of the maintainer. This International Standard provides task-steps which are examples of what to perform in order to implement the maintenance activities and tasks. The maintainer should ensure that the Maintenance Process exists and is functional prior to development of any software product. The Maintenance Process should be activated when a requirement exists to maintain a software product.

As soon as the process is activated, Maintenance Plans and Procedures should be developed and resources should be allocated specifically for maintenance. After the software product is delivered, the maintainers should modify the code and associated documentation in response to a modification request or problem report. The overall objective of software maintenance is to modify the existing product while preserving its integrity. This process supports the software product from its inception through migration to new environments, to its retirement. The process ends when the software product is finally retired.

The activities which comprise the Maintenance Process are:

a) Process Implementation.

b) Problem and Modification Analysis.

c) Modification Implementation.

d) Maintenance Review/Acceptance.

e) Migration.

f) Retirement.

Inputs are transformed or consumed by the maintenance activities to produce outputs. Controls provide guidance to ensure that the maintenance activity produces correct outputs. Outputs are the data or objects produced by the maintenance activity. Support identifies supporting life cycle processes of ISO/IEC 12207 used by the maintenance activities.

Figure 2 provides an overview of the Maintenance Process.

Copyright © 2006 IEEE. All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

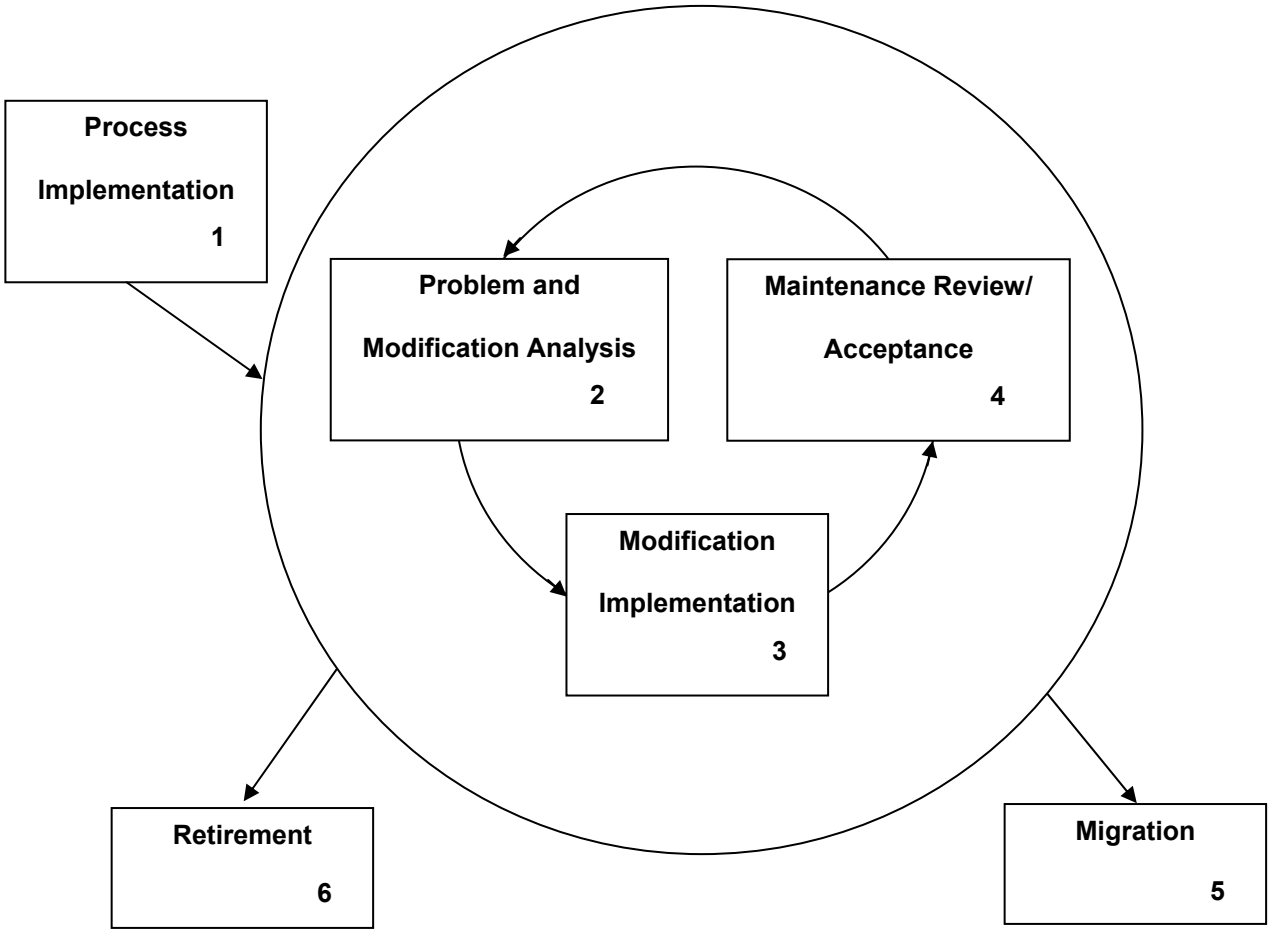



**Figure 2 — Maintenance Process**

## 5.1 Process Implementation

During Process Implementation, the maintainer establishes the plans and procedures which are to be executed during the Maintenance Process. The Maintenance Plan (see sub-clause 8.3.2 of this International Standard) should be developed in parallel with the Development Plan. The maintainer should also establish needed organizational interfaces during this activity.

### 5.1.1 Inputs

The inputs for the Process Implementation activity should include:

— Relevant baselines;

— System documentation, if available;

— A Modification Request (MR) or Problem Report (PR), if applicable.


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

### 5.1.2 Tasks

In order to effectively implement the Maintenance Process, the maintainer should develop and document a strategy for performing the maintenance. To accomplish this effort, the maintainer must execute the following tasks:

— Develop Maintenance Plans and Procedures;

— Establish MR/PR Procedures;

— Implement Configuration Management;

— Develop a Configuration Management Plan, (may not be necessary for MRs/PRs).

#### 5.1.2.1 Maintenance plans and procedures

The maintainer shall (ISO/IEC 12207 sub-clause 5.5.1.1) develop, document, and execute plans and procedures for conducting the activities and tasks of the Maintenance Process.

The Maintenance Plan should document the strategy to be used to maintain the system, while the Maintenance Procedures should provide a more detailed approach on how to actually accomplish the maintenance. In order to develop effective Maintenance Plans and Procedures, the maintainer should perform the following task-steps:

a) Assist the acquirer in developing the maintenance concept.

b) Assist the acquirer in determining the scope of maintenance.

c) Assist the acquirer in analyzing maintenance organization alternatives.

d) Ensure written designation as the maintainer for the software product.

e) Conduct resource analyses.

f) Estimate maintenance costs.

g) Perform a maintainability assessment of the system.

h) Determine transition requirements.

i) Determine transition milestones.

j) Identify the Maintenance Process which will be used.

k) Document the Maintenance Process in the form of operating procedures.

#### 5.1.2.2 MR/PR procedures

The maintainer shall (ISO/IEC 12207 sub-clause 5.5.1.2) establish procedures for receiving, recording, and tracking problem reports and modification requests from the users and providing feedback to the users. Whenever problems are encountered, they shall (ISO/IEC 12207 sub-clause 6.8) be recorded and entered into the Problem Resolution Process.

The maintainer should perform the following task-steps:

a) Develop an identification numbering scheme for MRs/PRs.

b) Develop a scheme for categorizing and prioritizing MRs/PRs.

--`,,```,,,,````-`-`,,`,,`,`,,`---
Copyright © 2006 IEEE. All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

c) Develop procedures for determining trend analysis.

d) Determine the procedures for an operator to submit a MR/PR.

e) Determine how initial feedback will be provided to the operators or users.

f) Determine how temporary work-arounds will be provided to the operators or users.

g) Determine how data is entered into the status accounting database.

h) Determine what follow-up feedback will be provided to the operators or users.

#### 5.1.2.3 Configuration management

| |
|---|
| The maintainer shall (ISO/IEC 12207 sub-clause 5.5.1.3) implement (or establish organizational interface with) the Configuration Management Process (ISO/IEC 12207 sub-clause 6.2) for managing modifications to the existing system. |

The maintainer needs to invoke the CM Process of ISO/IEC 12207.

### 5.1.3 Controls

Joint reviews (ISO/IEC 12207 sub-clause 6.6) should be used to control the outputs of the Process Implementation Activity.

### 5.1.4 Support

The Process Implementation activity uses the following supporting life cycle processes of ISO/IEC 12207:

— Documentation Process;

— Configuration Management Process;

— Quality Assurance Process;

— Joint Review Process.

### 5.1.5 Outputs

The outputs of this activity are:

— The Maintenance Plan;

— Training Plan;

— Maintenance Procedures;

— Project Management Plan;

— Problem Resolution Procedures;

— Measurement Plan;

— Maintenance Manual;

— Plans for User Feedback;


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

— The Transition Plan;

— Maintainability Assessment;

— Configuration Management Plan.

All outputs should be placed under configuration management.

## 5.2 Problem and Modification Analysis

This and subsequent activities are activated after the software transition and are called iteratively when the need for modification arises.

During the Problem and Modification Analysis Activity, the maintainer:

— Analyzes MRs/PRs;

— Replicates or verifies the problem;

— Develops options for implementing the modification;

— Documents the MR/PR, the results, and execution options;

— Obtains approval for the selected modification option.

Input for the Problem and Modification Analysis activity should be a validated Modification Request or Problem Report, system/project documentation, and requirements documentation.

### 5.2.1 Inputs

The inputs for the Problem and Modification Analysis activity should be:

— MR/PR;

— Baseline;

— Software repository;

— System documentation.

System Documentation includes:

— Configuration status information;

— Functional requirements;

— Interface requirements;

— Project planning data;

— Outputs from the Process Implementation Activity.

### 5.2.2 Tasks

Before modifying the system, the maintainer should analyze the MR/PR to determine its impact on the organization, the existing system, and the interfacing systems; develop and document recommended potential solutions; and obtain approval to implement the desired solution.

Copyright © 2006 IEEE. All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

#### 5.2.2.1 MR/PR analysis

The maintainer shall (ISO/IEC 12207 sub-clause 5.5.2.1) analyze the problem report or modification request for its impact on the organization, the existing system, and the interfacing systems for the following:

a) Type; for example, corrective, improvement, preventive, or adaptive to new environment;

b) Scope; for example, size of modification, cost involved, time to modify;

c) Criticality; for example, impact on performance, safety, or security.

In order to ensure that the requested MR/PR is feasible, the maintainer should perform the following task-steps:

a) Determine if the maintainer is adequately staffed to implement the proposed modification.

b) Determine if the program is adequately budgeted to implement the proposed modification.

c) Determine if sufficient resources are available and whether this modification will affect ongoing or projected projects (may not be necessary for PRs).

d) Determine the operational issues to be considered. For example, what are the anticipated modifications to system interface requirements, the expected useful life of the system, the operational priorities, safety, and security, security impacts, if it is not implemented? (may not be necessary for PRs).

e) Determine handling priority.

f) Classify the type of maintenance.

g) Determine the impact to current and future users.

h) Determine safety and security implications (may not be necessary for PRs).

i) Identify ripple effects.

j) Evaluate any software or hardware constraints that may result from the modifications.

k) Determine short-term and long-term costs (may not be necessary for PRs).

l) Determine the value of the benefit of making the modification.

m) Determine the impact on existing schedules.

n) Document any project or software risks resulting from the impact analysis.

o) Determine the level of test and evaluation required.

p) Determine the estimated management cost to implement the modification (may not be necessary for PRs).

q) Place developed artifacts under CM.

#### 5.2.2.2 Verification

The maintainer shall (ISO/IEC 12207 sub-clause 5.5.2.2) replicate or verify the problem.



--`,,```,,,,````-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

In order to ensure that the requested problem reports are valid, the maintainer should replicate or verify problems by performing the following task-steps:

a) Develop a test strategy to verify the problem.

b) Obtain affected software version from CM.

c) Install affected version.

d) Run test to verify problem, preferably with a copy of the affected data.

e) Document test results.

If the problem cannot be replicated for some reason, e.g., confidentiality of the data, other items such as organization rules, policies, documentation, should be checked.

NOTE—The verification task is not required for adaptive or perfective maintenance.

#### 5.2.2.3 Options

| |
|---|
| Based upon the analysis, the maintainer shall (ISO/IEC 12207 sub-clause 5.5.2.3) develop options for implementing the modification. |

The maintainer should perform the following task-steps:

a) Assign a work priority to the MR/PR.

b) Determine if a work-around exists for problems. If so, provide work-around to operator or user. (This task-step is not needed for adaptive or perfective maintenance.)

c) Define firm requirements for the modification.

d) Estimate the size and magnitude of the modification.

e) Develop different options to implement the modification.

f) Determine the impacts the options will have on the system hardware and users.

g) Perform a Risk Analysis for each of the options identified.

h) Record acceptance or rejection of the proposed option.

i) Develop an agreed-upon plan to implement the modification.

#### 5.2.2.4 Documentation

| |
|---|
| The maintainer shall (ISO/IEC 12207 sub-clause 5.5.2.4) document the problem/modification request, the analysis results, and implementation options. |

The following task-steps should be performed:

a) Verify that all appropriate analysis and project documentation has been updated. If none exists, develop documentation.

b) Review the proposed test strategy and schedule for accuracy.

c) Review resource estimates for accuracy.

Copyright © 2006 IEEE. All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

d) Update the status accounting database.

e) Include a Disposition Recommendation to indicate whether the MR/PR should be approved or disapproved.

#### 5.2.2.5 Approval

| |
|---|
| The maintainer shall (ISO/IEC 12207 sub-clause 5.5.2.5) obtain approval for the selected modification option as specified in the contract. |

Approval should also be obtained when maintenance is performed when agreements are not used to initiate maintenance. The maintainer may obtain this approval by performing the following task-steps:

a) Provide analysis results for approval by appropriate CM groups.

b) Participate at discussions regarding the modification.

c) Upon approval, update the status of the Modification Request.

d) Upon approval, update the requirements if the request is an enhancement (improvement).

### 5.2.3 Controls

Control is maintained through Joint reviews (ISO/IEC 12207 sub-clause 6.6).

At the end of this activity, a risk analysis should be performed. Using the output from the Maintenance Process Problem and Modification Analysis activity, the preliminary resource estimate should be revised, and a decision, that includes the user (customer) is made on whether to proceed to the Modification Implementation activity.

### 5.2.4 Support

The Problem and Modification Analysis activity uses the following supporting life cycle processes of ISO/IEC 12207:

— Documentation Process;

— Quality Assurance Process;

— Problem Resolution Process.

### 5.2.5 Outputs

The outputs of this activity are:

— Impact analysis;

— Recommended option;

— Approved modification;

— Updated documentation.

The impact analysis should include the following:

— Statement of the problem or new requirement;


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

— Problem or requirement evaluation;

— Classification of the type of maintenance required;

— Initial priority;

— Verification data (for corrective modifications);

— Initial estimate of resources required to modify the existing system.

Updated documentation should include:

— A Test Strategy;

— Updated test documentation, including test plan, test procedures and test reports;

— Software documentation;

— Updated requirements.

## 5.3 Modification Implementation

During the Modification Implementation Activity, the maintainer develops and tests the modification of the software product.

### 5.3.1 Inputs

The inputs to the Modification Implementation activity are:

— The Baseline;

— The Approved MR/PR;

— The Approved Modification Documentation.

The Baseline should include:

— System Architecture Definitions;

— The Modification Request Record;

— Source Code.

The Approved Modification Documentation should include:

— The Impact Analysis Report;

— Outputs from the Problem and Modification Analysis Activity.

### 5.3.2 Tasks

The maintainer performs analysis, then invokes the Development Process of ISO/IEC 12207 to effect the modification.

Copyright © 2006 IEEE. All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

#### 5.3.2.1 Analysis

The maintainer shall (ISO/IEC 12207 sub-clause 5.5.3.1) conduct analysis and determine which documentation, software units, and versions thereof need to be modified. These shall (ISO/IEC 12207 sub-clause 5.5.3.1) be documented.

The results of this additional analysis should be documented in the software documentation. This effort includes the following task-steps:

a) Identify the elements to be modified in the existing system.

b) Identify the interface elements affected by the modification.

c) Identify the documentation to be updated.

d) Update the software documentation.

#### 5.3.2.2 Development process

The maintainer shall (ISO/IEC 12207 sub-clause 5.5.3.2) enter the Development Process (ISO/IEC 12207 sub-clause 5.3) to implement the modifications. The requirements of the Development Process shall (ISO/IEC 12207 sub-clause 5.5.3.2) be supplemented as follows:

a) Test and evaluation criteria for testing and evaluating the modified and the un-modified parts (software units, components, and configuration items) of the system shall (ISO/IEC 12207 sub-clause 5.5.3.2 a) be defined and documented.

b) The complete and correct implementation of the new and modified requirements shall (ISO/IEC 12207 sub-clause 5.5.3.2 b) be ensured. It also shall (ISO/IEC 12207 sub-clause 5.5.3.2 b) be ensured that the original, unmodified requirements were not affected. The test results shall (ISO/IEC 12207 sub-clause 5.5.3.2 b) be documented.

The activities in the ISO/IEC 12207 Development Process should be tailored to meet the needs of the modification effort.

The Requirements Elicitation sub-process of ISO/IEC 12207 Amendment 1's Development Process is satisfied by the Process Implementation and Problem and Modification Analysis activities of the Maintenance Process.

### 5.3.3 Controls

Control of Modification Implementation should include Joint reviews (ISO/IEC 12207 sub-clause 6.6).

### 5.3.4 Support

The Modification Implementation activity uses the following supporting life cycle processes of ISO/IEC 12207:

— Documentation Process;

— Quality Assurance Process;


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

— Joint Review Process.

#### 5.3.5 Outputs

The outputs of this activity should include:

— Updated Test Plans and Procedures;

— Updated documentation;

— Modified source code;

— Test reporting;

— Measures.

The updated documentation should include:

— Updated Modification Records;

— Detailed Analysis Report;

— Updated requirements;

— Updated Test Plans, Test Procedures, and Test Reports;

— Updated training materials.

### 5.4 Maintenance Review/Acceptance

This activity ensures that the modifications to the system are correct and that they were accomplished in accordance with the approved standards using the correct methodology.

#### 5.4.1 Inputs

The inputs to the Maintenance Reviews/Acceptance activity are:

— The modified software;

— Modification test results.

#### 5.4.2 Tasks

Reviews are conducted to ensure that modifications are correct and approval is obtained for satisfactory completion of the modifications.

##### 5.4.2.1 Reviews

| |
|---|
| The maintainer shall (ISO/IEC 12207 sub-clause 5.5.4.1) conduct review(s) with the organization authorizing the modification to determine the integrity of the modified system. |

The following task-steps should be performed:

a) Trace the MR/PR from requirements, to design, to code.

b) Verify testability of the code.

Copyright © 2006 IEEE. All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

c) Verify conformance with coding standards.

d) Verify that only necessary software components were modified.

e) Verify that the new software components were integrated properly.

f) Check documentation to ensure that it was updated.

g) CM personnel build software items for testing.

h) Perform testing by an independent test organization.

i) Perform system tests on a fully integrated system.

j) Develop test report.

#### 5.4.2.2 Approval

| |
|---|
| The maintainer shall (ISO/IEC 12207 sub-clause 5.5.4.2) obtain approval for the satisfactory completion of the modification as specified in the contract. |

If maintenance was implemented without an agreement, approval should also be obtained. The following task-steps should be performed:

a) Obtain approval through the QA life cycle supporting process (ISO/IEC 12207).

b) Verify that the process has been followed.

c) CM prepares the delivery package and sends it to the operator's facility.

d) Conduct functional and physical configuration audits.

e) Notify the operators.

f) Perform installation and training at the operator's facility.

### 5.4.3 Controls

Control is exercised through Joint Reviews (ISO/IEC 12207 sub-clause 6.6).

### 5.4.4 Support

The Maintenance Review/Acceptance activity uses the following supporting life cycle processes of ISO/IEC 12207:

— Quality Assurance Process;

— Verification Process;

— Validation Process;

— Joint Review Process;

— Audit Process.


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

#### 5.4.5 Outputs

The outputs of this activity are:

— New Baseline, incorporating accepted modifications;

— Rejected modifications;

— An Acceptance report;

— Audit and Review Reports;

— A Software Qualification Test Report.

### 5.5 Migration

During a system's life, it may have to be modified to run in different environments. In order to migrate a system to a new environment, the maintainer needs to determine the actions needed to accomplish the migration, and then develop and document the steps required to effect the migration.

#### 5.5.1 Inputs

The inputs to the Migration activity are:

— The old environment;

— The new environment;

— The old baseline;

— The new baseline.

#### 5.5.2 Tasks

The Maintainer effects the migration by conforming with ISO/IEC 12207, developing a Migration Plan, notifying users of the migration, providing training, providing a notification of completion, assessing the impact of the new environment, and archiving data. All artifacts from the Migration activity are controlled by CM.

##### 5.5.2.1 Migration

| |
|---|
| If a system or software product (including data) is migrated from an old to a new operational environment, it shall (ISO/IEC 12207 sub-clause 5.5.5.1) be ensured that any software product or data produced or modified during migration are in accordance with ISO/IEC 12207. |

The following task-steps should be performed:

a) Identify all software products or data that were added or modified.

b) Verify that the tasks adhere to ISO/IEC 12207.

##### 5.5.2.2 Migration plan

| |
|---|
| A migration plan shall (ISO/IEC 12207 sub-clause 5.5.5.2) be developed, documented and executed. The planning activities shall (ISO/IEC 12207 sub-clause 5.5.5.2) include users. Items included in the plan shall (ISO/IEC 12207 sub-clause 5.5.5.2 ) include the following: |

Copyright © 2006 IEEE. All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

a) Requirements analysis and definition of migration;

b) Development of migration tools;

c) Conversion of software product and data;

d) Migration execution;

e) Migration verification;

f) Support for the old environment in the future.

The development of the Migration Plan should include input from the users. As part of this task, the maintainer should perform the following task-steps:

a) Analyze the migration requirements.

b) Determine the impact of migrating the software product.

c) Establish a schedule for performing the migration.

d) Identify data collection requirements for post-operation review.

e) Define and document the migration effort.

f) Determine and mitigate risks.

g) Identify needed migration tools.

h) Identify support for the old environment.

i) Develop and/or acquire migration tools.

j) Incrementally decompose software products and data for conversion.

k) Prioritize conversion of software products and data.

l) Convert software products and data.

m) Migrate software products and data to new environment.

n) Run parallel operations.

o) Verify migration through testing.

p) Provide support for old environment.

#### 5.5.2.3 Notification of intent

Users shall (ISO/IEC 12207 sub-clause 5.5.5.3) be given notification of the migration plans and activities. Notifications shall (ISO/IEC 12207 sub-clause 5.5.5.3) include the following:

a) Statement of why the old environment is no longer to be supported.

b) Description of the new environment with its date of availability.

c) Description of other support options available, if any, once support for the old environment has been removed.


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

The maintainer should also provide the users with the plan, procedures and the schedule. As part of this task, the maintainer should perform the following task-steps:

a) Identify all the locations which will be affected.

b) Process site feedback.

c) Identify site specific issues.

d) Promulgate the schedule.

#### 5.5.2.4 Implement operations and training

Parallel operations of the old and new environments may be conducted for smooth transition to the new environment (ISO/IEC 12207 sub-clause 5.5.5.4). During this period, necessary training shall (ISO/IEC 12207 sub-clause 5.5.5.4) be provided as specified in the contract.

As part of this task, the maintainer may perform the following task-steps for parallel operations:

a) Perform a site survey.

b) Install the equipment.

c) Install the software.

d) Perform preliminary tests to ensure a successful installation of the hardware and software.

e) Run the software under an operational load in parallel with the old system.

f) Collect data from the new and old products.

g) Perform data reduction and analysis.

The maintainer should perform the following task-steps for training:

a) Identify migration training requirements.

b) Schedule migration training requirements.

c) Conduct migration training review.

d) Update training plans.

#### 5.5.2.5 Notification of completion

When the scheduled migration arrives, notification shall (ISO/IEC 12207 sub-clause 5.5.5.5) be sent to all concerned. All associated old environment's documentation, logs, and code should be placed in archives (ISO/IEC 12207 sub-clause 5.5.5.5).

As part of this task, the maintainer should perform the following task-steps:

a) Promulgate changes to the migration schedule.

b) Document site specific issues and how they will be resolved.

c) Archive the old software and data.

Copyright © 2006 IEEE. All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

d) Remove the old equipment.

#### 5.5.2.6 Post-operation review

A post-operation review shall (ISO/IEC 12207 sub-clause 5.5.5.6) be performed to assess the impact of changing to the new environment. The results of the review shall (ISO/IEC 12207 sub-clause 5.5.5.6) be sent to the appropriate authorities for information, guidance, and action.

As part of this task, the maintainer should perform the following task-steps:

a) Review the results of operating the systems in parallel.

b) Identify potential risk areas.

c) Identify site specific issues.

d) Document any lessons learned.

e) Generate and forward an Impact Analysis report.

#### 5.5.2.7 Data archival

Data used by or associated with the old environment shall (ISO/IEC 12207 sub-clause 5.5.5.7) be accessible in accordance with the contract requirements for data protection and audit applicable to the data.

As part of this task, the maintainer should perform the following task-steps:

a) Store the old software and data.

b) Make copies of the old software and data.

c) Store the media in a safe place.

### 5.5.3 Controls

Control is exercised through Joint reviews (ISO/IEC 12207 sub-clause 6.6).

### 5.5.4 Support

The Migration activity uses the following supporting life cycle processes of ISO/IEC 12207:

— Documentation Process;

— Configuration Management Process;

— Quality Assurance Process;

— Verification Process;

— Validation Process;

— Joint Review Process;

— Audit Process;


--`,,```,,,,````-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

— Problem Resolution Process.

### 5.5.5 Outputs

The outputs of this activity are:

— Migration Plan;

— Migration tools;

— Notification of Intent;

— Migrated Software Product;

— Notification of Completion;

— Measures;

— Archived data.

## 5.6 Software retirement

Once a software product has reached the end of its useful life, it must be retired. An analysis should be performed to assist in making the decision to retire a software product. The analysis is often economic-based and may be included in the Retirement Plan. Analysis should determine if it is cost effective to:

— Retain outdated technology;

— Shift to new technology by developing a new software product;

— Develop a new software product to achieve modularity;

— Develop a new software product to facilitate maintenance;

— Develop a new software product to achieve standardization;

— Develop a new software product to facilitate vendor independence.

The software product may be replaced by a new software product but in some cases it will not be replaced. In order to retire a software product, the maintainer should determine the actions needed to accomplish the retirement, and then develop and document the steps required to effect the retirement. Consideration should be given to accessing data stored by the retired software product.

All artifacts from the Retirement activity are controlled by CM.

### 5.6.1 Inputs

The inputs to the Retirement Activity are:

— The old software product baseline to be retired;

— The new software product;

— The old environment.

Copyright © 2006 IEEE. All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

### 5.6.2 Tasks

The Maintainer effects the retirement by conforming with ISO/IEC 12207, developing a Retirement Plan, notifying users of the retirement, implementing parallel operations and training, providing a notification of completion, and archiving data. All artifacts from the Retirement activity are controlled by CM.

#### 5.6.2.1 Retirement plan

A retirement plan to remove active support by the operation and maintenance organizations shall (ISO/IEC 12207 sub-clause 5.5.6.1) be developed and documented. The planning activities shall (ISO/IEC 12207 sub-clause 5.5.6.1) include users. The plan shall (ISO/IEC 12207 sub-clause 5.5.6.1) address the items listed below. The plan shall (ISO/IEC 12207 sub-clause 5.5.6.1) be executed.

a) Cessation of full or partial support after a certain period of time;

b) Archiving of the software product and its associated documentation;

c) Responsibility for any future residual support issues;

d) Transition to the new software product, if applicable;

e) Accessibility of archive copies of data.

As part of this task, the maintainer should perform the following task-steps:

a) Analyze the retirement requirements.

b) Determine the impact of retiring the software product.

c) Identify replacement software product, if any.

d) Establish a schedule for retiring the software product.

e) Identify the responsibility for any future residual support.

f) Define and document the retirement effort.

#### 5.6.2.2 Notification of intent

Users shall (ISO/IEC 12207 sub-clause 5.5.6.2) be given notification of the retirement plans and activities. Notifications shall (ISO/IEC 12207 sub-clause 5.5.6.2) include the following:

a) Description of the replacement or upgrade with its date of availability;

b) Statement of why the support product is no longer to be supported;

c) Description of other support options available, once support has been removed.

As part of this task, the maintainer should perform the following task-steps:

a) Identify all the locations which will be affected.

b) Identify site specific issues.


--`,,```,,,,````-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

c) Promulgate the schedule.

d) Process site feedback.

#### 5.6.2.3 Implement parallel operations and training

Parallel operations of the retiring and the new software product should be conducted for smooth transition to the new system (ISO/IEC 12207 sub-clause 5.5.6.3). During this period, user training shall (ISO/IEC 12207 sub-clause 5.5.6.3) be provided as specified in the contract.

As part of this task, the maintainer should perform the following task-steps:

a) Perform a site survey.

b) Install the equipment.

c) Install the software product.

d) Perform preliminary tests to ensure a successful installation of the hardware and software.

e) Run the software product under an operational load in parallel with the old system.

f) Collect data from the new and old products.

g) Perform data reduction and analysis.

#### 5.6.2.4 Notification of completion

When the scheduled retirement arrives, notification shall (ISO/IEC 12207 sub-clause 5.5.6.4) be sent to all concerned. All associated development documentation, logs, and code should be placed in archives, when appropriate (ISO/IEC 12207 sub-clause 5.5.6.4).

As part of this task, the maintainer should perform the following task-steps:

a) Promulgate changes to the retirement schedule.

b) Document site specific issues and how they will be resolved.

c) Archive the old software and data.

d) Remove the old equipment.

#### 5.6.2.5 Data archival

Data used by or associated with the retired software product shall (ISO/IEC 12207 sub-clause 5.5.6.5) be accessible in accordance with the contract requirements for data protection and audit applicable to the data.

Consideration should be given to updating archive media to CD-ROM and other digital disk products to simplify retrieval. As part of this task, the maintainer should perform the following task-steps:

a) Store the old software and data obtained during the retirement activity.

b) Make copies of the old software and data obtained during the retirement activity.

c) Store the media in a safe place.

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright © 2006 IEEE. All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

### 5.6.3 Controls

Control is exercised through Joint reviews (ISO/IEC 12207 sub-clause 6.6).

### 5.6.4 Support

The Software Retirement activity uses the following supporting life cycle processes of ISO/IEC 12207:

— Documentation Process;

— Configuration Management Process;

— Quality Assurance Process;

— Joint Review Process;

— Audit Process.

### 5.6.5 Outputs

The outputs of this activity are:

— A Retirement Plan;

— A Notification of Intent;

— Retirement results;

— Trained people;

— A retired software product;

— A Notification of Completion;

— Measures;

— Archived baseline and data.

# 6 Execution considerations

## 6.1 Introduction

The software maintenance life cycle process begins with Process Implementation where planning for maintenance is performed and ends with the retirement of the software product. It includes modification of code and documentation due to a problem or need for improvement. The objective of the Maintenance Process is to modify an existing software product while preserving its integrity. The following provides execution considerations.

The Maintenance Process is needed because the operational environment detects errors and because it introduces the need for unforeseen new and/or modified capability. If the software product is developed using Computer-Aided Software Engineering (CASE) tools, maintenance is still needed. CASE tools facilitate maintenance but do not eliminate the requirement for maintenance. If no application code is developed, i.e., the software product consists solely of off-the-shelf products, maintenance may still be required. Maintenance of off-the-shelf software products by the acquirer or supplier will usually involve modification of the interfaces, both data and operational, to the product.

--`,,```,,,,````-`-`,,`,,`,`,,`---



Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

Consideration should be given to implicit requirements and constraints imposed on the original developer. Circumstances may have changed and some of the original requirements may no longer be applicable.

During execution of the Development, Operations, and Maintenance Processes of ISO/IEC 12207, any problems detected are recorded and monitored by the Problem Resolution process of ISO/IEC 12207. Modification Requests (MRs) or Problem Reports (PRs) are submitted. Often, these are referred to as change requests. The Problem Resolution process of ISO/IEC 12207 analyzes and resolves problems. It also determines if an MR/PR is a problem or an enhancement. The Configuration Management (CM) process of ISO/IEC 12207 records and reports the status of MRs/ PRs. The Configuration Control activity of the CM process then decides whether to approve the request. Approved MRs/PRs are then implemented by calling the Development Process. Some systems contain software. Systems engineering aspects are discussed in ISO/IEC 15288.

Maintenance is needed to ensure that the software product continues to satisfy the user requirements. Maintenance is applicable to software developed using any development life cycle model (e.g., incremental, waterfall, evolutionary).

Constraints imposed by the operational environment impact the Maintenance Process. Often there are 24 hour non-stop operations/maintenance service environments. Software maintenance needs to be performed on systems that cannot be stopped easily. Maintenance strategies must be put in place for this type. Maintenance to such software must be carefully planned in order not to degrade the agreed upon service level. The maintainer should always be prepared in case maintenance action causes a general system failure.

The Maintenance Process may consume a significant portion of life cycle costs. Analysis of the types of maintenance performed helps to provide an understanding of the costs.

## 6.2 Types of maintenance

Corrective maintenance refers to modifications necessitated by actual errors in a software product. If the software product does not meet its requirements, corrective maintenance is performed. Emergency maintenance is an unscheduled modification performed to temporarily keep a system operational pending corrective maintenance.

Preventive Maintenance refers to the modifications necessitated by detecting potential errors in a software product.

Adaptive and Perfective Maintenance refers to modifications that are enhancements to a software product. These modifications are those that were not in the design specifications or the released software. Adaptive modifications are those modifications necessary to accommodate a changing environment. Adaptive modifications include modifications to implement new system interface requirements, new system requirements, or new hardware requirements. Perfective modifications improve the software product's performance or maintainability. A perfective modification might entail providing new functionality improvements for users or reverse engineering to create maintenance documentation that did not exist previously or to modify existing documentation.

Software maintenance requires modifications to an existing structure or system, i.e., software modifications are introduced into an existing architecture and must allow for constraints imposed by the design structure. Thus, enhancements in the form of adaptive and perfective maintenance, are often very costly and time consuming. Enhancements may consume a significant portion of maintenance costs.

## 6.3 Arrangements for maintenance

The acquirer may enter into an agreement with the original developer to perform maintenance or a separate third party may be the maintainer. Maintenance can also be provided by internal two party agreements.

Copyright © 2006 IEEE. All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

Maintenance service models should be agreed upon. The models should address all types of maintenance and include new development unless the amount of the costs and resources exceed the initial fixed price. Someone should identify what types of maintenance are included. Then a comprehensive contract should be made with a fixed price. Suggested types are: Type 1 - Blanket contract with fixed amounts for maintenance. It includes all types of maintenance and may include new development; Type 2 - Split contract for maintenance. It typically includes corrective maintenance for an agreed period. Preventive, adaptive, and perfective maintenance are usually contracted separately for each.

ISO/IEC 12207 provides detailed tasks for the derivation of an agreement between the acquirer and supplier. This should be used to aid the derivation of a maintenance agreement whether acquirer or supplier are from the same or different organizations. These agreements are often referred to as Service Level Agreements. Specific maintenance issues are discussed later.

If the acquirer requires software maintenance from the developer after delivery, or at the end of a warranty period, this should be stipulated in the agreement. Updated documentation should be stipulated in the agreement as a deliverable. Training should also be stipulated. The supplier should then prepare procedures for the maintenance task, keep these procedures up to date and check that the activities comply with the agreement requirements and prepared procedures. Empirical data suggests that the use of procedures is cost effective. The items to be maintained, the maintenance procedures, and the time for which they are to be maintained, should be specified in the maintenance plan.

The supplier (the maintainer) and the acquirer should first agree on a maintenance agreement and stipulate procedures to incorporate modifications into the maintained software products. Similar procedures should be used by the original developer and third party maintainers.

These procedures should include:

— Basic rules used to determine when the software can be locally corrected or when a new baseline, using the Development Process of ISO/IEC 12207 for installation and release, is required;

— Descriptions of types of releases depending on their frequency or their affects on software operation (e.g., emergency releases, periodic releases);

— Ways in which the acquirer will be informed on the status of current or future modifications;

— Methods to confirm that the changes will not introduce other problems into the software;

— Classification of modifications according to major/minor or other distinction which will dictate how a modification is authorized, processed, and approved.

## 6.4 Tools for maintenance

A potential means of containing software maintenance costs is to use CASE tools. These tools aid software maintenance activities. The vision for CASE is an interrelated set of tools supporting all aspects of software development and maintenance [ISO/IEC TR 14471]. This interrelated collection of CASE tools should be brought together in the form of a Software Engineering Environment (SEE) to support the methods, policies, guidelines, and standards that support software maintenance activities. A Software Test Environment (STE) should also be provided for the maintainer so that the modified software product can be tested in a non-operational environment. The SEE provides the tools to initially develop and modify the software products. The STE provides the test environment. The STE should be used to test the modified software products in a non-operational environment.

To date the adoption of CASE tools has met with limited success. Maintainers should plan these efforts carefully [ISO/IEC TR 14471]. ISO/IEC 19759 provides additional information regarding tools.


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

## 6.5 Software maintenance measurement

Software quality is an important consideration in the maintenance of a software product. Maintainers should have a software quality program that includes the six characteristics of software quality described in ISO/IEC 9126-1. The measurement process from ISO/IEC 12207 Amd 1 (F.3.1.6) should be implemented to identify, define, select, apply, validate, and improve software measurement for software maintenance. ISO/IEC 15939 provides additional measurement information.

As part of software measurement, the maintainer should determine the effort (in terms of resources expended) for corrective, preventive, adaptive, and perfective maintenance. Data should be collected, analyzed, and interpreted in order to facilitate Maintenance Process improvement and to obtain a better understanding of where maintenance costs are being expended. Empirical measurement data should be collected in order to assist life cycle cost estimating and where appropriate to explain the quality issues of a software product and to explain what could be done to improve the developers' processes for the next software product.

## 6.6 Documentation of process

The detailed software Maintenance Process (clause 6 of this International Standard) should be documented so that all maintenance personnel follow the same process. The measures should support the process and related software process improvement efforts.

## 6.7 Early involvement in development

Data suggests that the cost of software maintenance and the maintainer's ability to conduct software maintenance is greatly influenced by what occurs or does not occur during the software development process. In many cases, the maintainer can not be involved due to contractual or other reasons. Specifically, when maintenance is outsourced to a third party, there is often no opportunity for involvement. When the maintainer can be involved during development, the maintainer should be involved.

Functions performed by the maintainer should include:

— Plan for the logistics of supporting the software product;

— Plan for knowledge transfer;

— Ensure the maintainability of the software product;

— Support the planning for the transition of software products from development to maintenance.

Planning is discussed in detail in sub-clause 8.3 of this International Standard. Early involvement in projects by the maintainer can help in stating, establishing and clarifying the maintainability requirements of the software. ISO/IEC 9126 should be used to explicitly define maintainability and other software quality characteristics. Maintainability can be improved by maintainer participation in the quality assurance, verification, and validation supporting life cycle processes of ISO/IEC 12207. The maintainer should:

— Participate in reviews;

— Perform code analysis;

— Trace requirements;

— Perform Verification and Validation.

Copyright © 2006 IEEE. All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

## 6.8 Maintainability

Software maintainability and maintenance are important aspects of dependability. Maintainability is an important feature of software for the acquirer, supplier, and user. Maintainability requirements should be included in the Initiation Activity of the Acquisition Process of ISO/IEC 12207 and should be evaluated throughout the Development Process of ISO/IEC 12207. Variations in design should be monitored throughout development for impact to maintainability. Various measures, including complexity measures, should be used to define and assess the quality of software. Both qualitative and quantitative evaluation is important. There are four maintainability sub-characteristics that address analysability, changeability, stability and testability of software. These four dimensions affect the effort (not speed) and ease of software modifications.

### 6.8.1 Maintainability and the development process

Maintainability requirements, which are non-functional requirements, should be developed and agreed upon early in a software project. When software is acquired from a third party, agreement on the level of maintainability required should be developed between the acquirer and supplier as part of the Initiation Activity of ISO/IEC 12207.

The capability to monitor and evaluate maintainability criteria (or objectives) identified for each requirement should be developed during software development. The capability describes qualitative and quantitative software maintainability requirements specified by the customer. It defines the criteria and the ways of checking them. Qualitative requirements (e.g., usability, maintainability) are used to define the techniques employed to facilitate maintenance costing and resources. Quantitative requirements are used to define maintainability ratings, rating levels, or quality criteria and the measures used to determine values or indicators throughout the various software life cycle phases.

The maintainability sub-characteristics that have been specified should be reviewed and controlled during software development. An estimated effort to ensure the verification of maintainability by the maintainer should be stated in the maintenance strategy document. Developers should implement requirements for maintainability and maintainers should monitor execution. The effort should be part of the software maintenance strategy.

One of the key factors in applying ISO/IEC 12207 is the development of a software maintenance strategy (ISO/IEC TR 15271, Guide for ISO/IEC 12207). Accordingly, a maintenance strategy should be developed and maintenance should be planned (clause 8 of this International Standard).

A software maintenance strategy should also be established prior to design. Early maintainer involvement in a software project has the potential to reduce maintenance costs. There are many actions, including software maintenance planning, to be performed during the development process. These actions should be documented in the maintenance plan (sub-clause 8.3.2 of this International Standard).

Maintainability is affected by the architecture, design, the coding and its programming language and the testing activities. ISO/IEC 19759 provides additional information regarding good architecture and design approaches that help maintainability.

The following aspects, all of which affect maintainability, should be taken into account in choosing the programming language:

— Language portability;

— Language legibility;

— Language stability;

— Self-documentation;

--`,,```,,,,````-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

— Tolerance of programming "tricks" which reduce program clarity;

— Program structuring possibilities;

— The ease with which new releases can be produced;

— Data structuring possibilities;

— Availability of a compiler and other such tools;

— Stability of a compiler and other such tools;

— Test possibilities during compiling and runs;

— The availability of software engineering and software test environments to assist in production, debugging, configuration management and the satisfaction of reliability and quality requirements;

— How long the various development tools will be viable.

### 6.8.2 Maintainability and specific activities in the development process

#### 6.8.2.1 Software requirements analysis

The software specifications should exhaustively and unambiguously describe the maintainability requirements of the software. These should be included in the quality characteristics specifications required by ISO/IEC 12207. The following aspects affect maintainability and should be considered:

— The identification and definition of functions, particularly optional functions;

— The accuracy and logical organization of data;

— Interfaces (machine and users), particularly future interfaces;

— The performance requirements, including the effects of any corrections and additions;

— Requirements imposed by the planned environment including scalability requirements and projected system growth;

— The granularity of requirements as it impacts the ease or difficulty of traceability;

— The Software Quality Assurance Plan which should emphasize documentation and its compliance.

#### 6.8.2.2 Software architectural design

This activity transforms the requirements for the software item into an architecture that describes its top-level structure and identifies its software components (ISO/IEC 12207). An example of this architecture is when a user application layer and infrastructural system operating layer are appropriately independent for an internet type system, such architecture strongly helps to separately maintain application from infrastructural software. The main features of this Development Process Activity of ISO/IEC 12207 which affect maintainability are the choice of the program structure, the breakdown into entities and the flow of data through them. As in other activities, it is important to use the data processing knowledge of the programming teams since this can, in particular, reveal possibilities of using parts of existing programs or libraries which have already proven their dependability.

Copyright © 2006 IEEE. All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

#### 6.8.2.3 Software detailed design

This Development Process Activity of ISO/IEC 12207 provides a detailed design for each software component, for interfaces, and databases. The activity produces an accurate, detailed description of each function to complete the proposed programming solution.

#### 6.8.2.4 Software coding and testing

This Development Process Activity of ISO/IEC 12207 develops, documents, and tests software units, and databases. Software maintainability will be improved by upgrading the quality of documentation. Quality documentation should provide information which can help in performing the Maintenance Process. Suggestions for improving maintainability include:

— Ensuring legibility;

— Pursuing structured code;

— Reducing code complexity;

— Providing accurate code comments;

— Using indentation and white space;

— Eliminating classic traps by considering the language weaknesses;

— Using techniques to facilitate error-tracing;

— Ensuring traceability of source code to design;

— Using coding standards;

— Reducing complexity of decisions and control flow;

— Conducting inspections of code and test cases;

— Maintaining documentation during the development cycle.

#### 6.8.2.5 Software qualification testing

This activity ensures that each implemented software item is tested for compliance with its qualification requirements. The test cases used during software development should be kept for regression testing after modification. In addition, the development history of a program should be available for maintenance in order to better understand the evolution of the software during development.

### 6.9 Software transition

Software transition is a controlled and coordinated sequence of actions wherein software development passes from the organization performing initial software development to the organization performing software maintenance. If maintenance responsibility will transfer from one organization to another, a Transition Plan should be developed. The plan should address:

— The transfer of hardware, software, data, support services, and experience from the developer to the maintainer;

— The tasks needed for the maintainer to implement the software maintenance strategy (e.g., staffing, training, installation, replicating maintenance problems);


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

— Assessing knowledge transfer and documentation;

— Outstanding problems and new requirements by priority;

— Assessing the test environment readiness;

— Transfer of as-built configuration information for the source code and object code, including open or deferred problem reports and new requirements, number and location of media masters which may need to be updated during maintenance.

### 6.10 Documentation

Maintainers are often faced with providing maintenance for a software product for which little or no documentation exists. When faced with this situation, the maintainer should create the needed documentation during the planning and before the final transition. Maintainers should create the needed documentation while performing the following tasks in order to prepare for maintenance:

a) Understand the problem domain (the type of application). Read any documentation (if available), discuss the software product with developers (if available), and operate the software product.

b) Learn the structure and organization (e.g., control flow, data flow, data structures, call graph) of the software product. Inventory the software product, place the software product under CM, rebuild the software product from the CM libraries, produce call trees, and analyze the structure of the software product.

c) Determine what the software product is doing. Review specifications (if available), review overall structure, analyze call trees, read the code, provide oral presentations to other maintainers, and add comments to code.

d) Gradually build confidence by addressing low risk modifications and progressively more risky and complex modifications to maintain the stability of the software.

Maintainers should document the software product as the guidance listed above is executed. Documents such as specifications, programmers' maintenance manuals, users manuals, and installation guides, should be updated or created as necessary.

There are various factors that influence documentation creation/updating in the maintenance environment. Some factors include: access to source code, availability of tools to analyze the code, ability to operate the software product to determine capabilities, and availability of a Software Test Environment (STE).

## 7 Software maintenance strategy

### 7.1 Introduction

This sub-clause discusses development of the software maintenance strategy. The strategy prepares for the human and materiel resources required to provide software maintenance for software products. Maintainers should monitor the development effort for maintainability. Results from maintainability analyses should be used as aids in planning for maintenance. This analysis should be provided as input into development of the maintenance strategy. The software maintenance strategy should consist of the following elements:

— Maintenance concept;

— Maintenance plan;

— Resource analysis.

Copyright © 2006 IEEE. All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

## 7.2 The maintenance concept

Determining the Maintenance Concept should be the first step in developing the software maintenance strategy. To achieve optimum software maintenance planning, the maintenance concept should be developed when the initial software product needs are first expressed.

The maintenance concept should address:

— The scope of software maintenance;

— Defining the overall Maintenance Process;

— The designation of who will provide maintenance;

— An estimate of maintenance costs.

NOTE—The Maintenance Concept is documented in the Maintenance Plan.

### 7.2.1 Scope

Scope relates to how responsive the maintainer will be. It should define how much support the maintainer will provide. Budgetary constraints often dictate the scope of maintenance. The scope of maintenance should address:

— Types of maintenance to be performed;

— Level of documentation to be maintained;

— Responsiveness;

— Level of training to be provided;

— Delivery support;

— Help desk support.

### 7.2.2 Defining the process

The Maintenance Concept should include an overview of the process to be used for post-delivery software maintenance. The process overview should identify high level tasks. An early attempt should be made to identify the various organizations involved with each of the software maintenance tasks.

### 7.2.3 Designation of who will provide maintenance

Designation of who will provide maintenance is an important issue and should be addressed early and documented in the Maintenance Concept. This is equally applicable for in-house efforts. For outsourced third party agreement maintenance efforts, the Maintenance Concept should note that maintenance will be outsourced. The primary processes of Acquisition and Supply of ISO/IEC 12207 provide details regarding acquiring and supplying software services.

The designation of the maintainer should be based on a number of factors including:

— Service level required;

— Life of the software product;

— Long term costs;

--`,,```,,,,````-`-`,,`,,`,`,,`---



Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

— Startup costs;

— Availability of space;

— Qualifications;

— Availability;

— Schedule;

— Domain knowledge.

#### 7.2.4 Estimate of maintenance costs

An estimate of maintenance costs should be prepared. The costs should be a function of the scope of maintenance. Additional factors that should be included are:

— Travel to user locations;

— Training for maintainers as well as users;

— Cost and annual maintenance for the Software Engineering Environment and the Software Test Environment;

— Personnel costs such as salaries and benefits.

When the Maintenance Concept is developed, the costs should be estimated based on the limited data available, including the cost of system downtime. As the development effort progresses, the estimates should be refined. Historical measurement data should be used as input to estimating maintenance costs.

### 7.3 Maintenance planning

#### 7.3.1 Introduction

The purpose of Maintenance Planning is to plan maintenance activities and to acquire the resources required sufficiently early for them to be available as soon as the software product transitions to maintenance. Planning is initiated once the software maintenance concept has been defined and culminates in a maintenance plan used to guide maintainers once the software enters service. IEEE Std 1058 may be used for guidance in maintenance planning.

#### 7.3.2 The maintenance plan

The planning of maintenance activities and tasks should begin as soon as the Maintenance Concept, described above, has been defined. It culminates in the preparation of a Maintenance Plan. The Maintenance Plan should be prepared during software development by the maintainer and should include how users will request modifications to the software product.

The Maintenance Plan should cover:

— Why maintenance will be needed;

— Who will do what work;

— What the roles and responsibilities of everyone involved will be;

— How the work will be performed;

Copyright © 2006 IEEE. All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

— What resources will be available for maintenance;

— Where maintenance will be performed;

— When maintenance will commence;

— System description;

— Agreement protocols;

— Training;

— Controls;

— Records and reports.

### 7.3.3 Maintenance plan topics

This sub-clause provides guidelines for developing a Maintenance Plan. Topics for inclusion in a Maintenance Plan are suggested. Based on the size of the effort, a determination should be made as to which topics should be included. Some information can be described by referencing another document.

a) Introduction

   1) Describe the system to be supported

   2) Identify the initial status of the software

   3) Describe why support is needed

   4) Identify the maintainer/support organization

   5) Identify the specific software processes covered by the maintenance effort

   6) Describe any agreement protocols between customer and supplier

   7) Identify where maintenance will be performed

   8) Identify when maintenance will commence

   9) Identify costs to provide maintenance

   10) Identify the schedule

b) Identification and Control of the plan

   1) Identify the date of issue

   2) Identify the status of the plan

   3) Identify the issuing organization

   4) Identify the approval authority

   5) Describe the modification procedure for the plan

   6) Insert a modification history section

   7) Insert a glossary


--`,,```,,,,````-`-`,,`,,`,`,,`---
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

c) References (to higher level policies, procedures, and documents and to lower level plans and procedures providing additional details)

1) Identify the documents placing constraints on the maintenance effort

2) Identify documents referenced by the maintenance plan

3) Identify any supporting documents supplementing or implementing the maintenance plan

d) Definitions

1) Define or reference all terms required to understand the maintenance plan

2) Describe all abbreviations and notations used

e) Maintenance concept

1) Describe the concept, including the level of support for the system (e.g., only implementing corrective maintenance)

2) Identify the support period

f) Organization and maintenance activities

1) Pre-Delivery roles and responsibilities of the maintainer

i) Process Implementation

ii) Establish Infrastructure

iii) Establish Human Resource Process

iv) Establish the Software Maintenance Process

v) Develop the maintainability plan

vi) Monitor development execution for maintainability

vii) Develop the transition plan

viii) Participation by the maintainer in development activities

ix) Interface with other organizations

2) Post-Delivery roles and responsibilities of the maintainer

i) Process Implementation

ii) Problem and Modification Analysis

iii) Modification Implementation

iv) Maintenance Review/Acceptance

v) Migration

vi) Retirement

vii) Problem Resolution (includes Help Desk)

Copyright © 2006 IEEE. All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

viii) Train personnel (maintainer and user), as applicable

ix) Improve the process

x) Factors that determine organizational maintenance priorities

xi) The process for assigning a priority to a work package

xii) How resources are assigned to prioritized work packages

xiii) The schedule estimating method

xiv) Interface with other organizations

3) Role of the operator

i) Acceptance testing

ii) Interface with other organizations

g) Resources

1) Personnel

i) Size of staff for the project

2) Software

i) Identify software needed to support system (includes system plus SEE/STE/tools requirements)

3) Hardware

i) Identify hardware needed to support system (includes system plus SEE/STE requirements)

4) Facilities

i) Identify facilities requirements

5) Special procedural requirements (e.g., security, access rights, and documentation control)

6) Cost estimating

i) Describe the cost estimating method

7) Documentation

i) Software Quality Plan

ii) Project Management Plan

iii) Configuration Management Plan

iv) Measurement Plan

v) Development documents

vi) Maintenance manuals

--`,,```,,,,````-`-`,,`,,`,`,,`---



Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

vii) Verification Plan

viii) Validation Plan

ix) Test Plan, Test Procedures, and Test Reports

x) Training Plan

xi) User's Manual(s)

8) Data management

i) Identify repositories

9) Other resource requirements (if needed)

h) Process (how the work will be performed)

1) Maintainer's process (give an overview of the process, do not spell out the entire process in the maintenance plan)

2) Defined process (identify actions to be performed for each activity in the process)

i) Training

1) Identify training needs of the Maintainer and Users

j) Software maintenance control requirements

1) Describe the deviation policy

2) Describe control procedures

3) Identify quality control measures

4) Describe standards, practices, and conventions

5) Identify risks

k) Maintenance records and reports

1) Describe how information will be collected and provided

2) Lists of requests for assistance, modification requests, or problem reports

3) Status of requests by categories

4) Priorities of requests

5) Measurement data to be collected on maintenance activities

## 7.4 Resource analysis

The last element of the software maintenance strategy is resource analysis. Once the scope of maintenance and which organization will perform maintenance is known, the personnel, maintenance environment, and financial resource requirements can be determined. The acquirer, with assistance from the supplier (developer) normally determines the resource requirements for software maintenance. Personnel, environment, and financial resources should be addressed.

--`,,```,,,,````-`-`,,`,,`,`,,`---
Copyright © 2006 IEEE. All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

### 7.4.1 Personnel resources

Resource requirements pose a major challenge in software maintenance planning. Personnel requirements are a major cost factor, and, at the same time, the most difficult factor to accurately estimate. The two most popular approaches for estimating resources for software maintenance are use of parametric models and use of experience. Empirical and historical data are commonly used for both approaches and are required for parametric modeling.

It is suggested that a standard, agreed-upon methodology for estimating maintenance, be used. A separate maintenance staffing study, which addresses the methodology for determining personnel resources and the results, should be developed.

### 7.4.2 Environment resources

Software development and maintenance are specialized activities and need separate systems dedicated to them. Separate Software Engineering Environments (SEEs) and Software Test Environments (STEs) are suggested. The maintainer should assist the acquirer with the plan for the maintenance environment. It is critical to get the maintenance environment included in early planning efforts when funds are allocated and a budget is determined for the development and maintenance of the software product.

### 7.4.3 Financial resources

The third and final aspect of resources are the financial resources. To provide effective maintenance support, the Maintainer should have a budget which addresses the following:

— Salaries;

— Training (2-3 weeks per year per person);

— Annual maintenance costs for software licenses;

— Travel;

— Technical publications;

— Hardware and software for the engineering and test environments;

— Upgrades to the hardware and software for the engineering and test environments.


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

# Annex A
(informative)

## Cross-reference between ISO/IEC/IEEE 14764 and ISO/IEC 12207 and ISO/IEC 12207 Amd 1

| Clause in ISO/IEC/IEEE 14764 | Clause in ISO/IEC 12207 | Clause in ISO/IEC 12207, Amd 1 |
|---|---|---|
| 1.2 | 1.2 | |
| 4.2 | 3.5 | |
| 5.1 | 4.1.1.1/4.1.1.2/ 4.1.1.3 | F.3.4/F.3.4.2 |
| 6 | 5.5 | |
| 6.1 | 5.5.1 | |
| 6.1.2.1 | 5.5.1.1 | |
| 6.1.2.2 | 5.5.1.2 | |
| 6.1.2.3 | 5.5.1.3 | |
| 6.1.3 | 6.6 | |
| 6.2 | 5.5.2 | |
| 6.2.2.1 | 5.5.2.1 | F.1.5 |
| 6.2.2.2 | 5.5.2.2 | |
| 6.2.2.3 | 5.5.2.3 | |
| 6.2.2.4 | 5.5.2.4 | F.1.5 |
| 6.2.2.5 | 5.5.2.5 | |
| 6.2.3 | 6.6 | |
| 6.3 | 5.5.3 | |
| 6.3.2.1 | 5.5.3.1 | |
| 6.3.2.2 | 5.3/5.5.3.2 | F.1.3/F.1.3.1-F.1.3.11 |
| 6.3.3 | 6.6 | |
| 6.4 | 5.5.4 | |

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright © 2006 IEEE. All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

| Clause in ISO/IEC/IEEE 14764 | Clause in ISO/IEC 12207 | Clause in ISO/IEC 12207, Amd 1 |
|---|---|---|
| 6.4.2.1 | 5.5.4.1 | |
| 6.4.2.2 | 5.5.4.2 | F.3.4 |
| 6.4.3 | 6.6 | |
| 6.5 | 5.5.5 | |
| 6.5.2 | 5.5.5.4 | F.3.4 |
| 6.5.2.1 | 5.5.5.1 | |
| 6.5.2.2 | 5.5.5.2 | F.1.5 |
| 6.5.2.3 | 5.5.5.3 | |
| 6.5.2.4 | 5.5.5.4 | F.1.5/F.3.4 |
| 6.5.2.5 | 5.5.5.5 | |
| 6.5.2.6 | 5.5.5.6 | |
| 6.5.2.7 | 5.5.5.7 | |
| 6.5.3 | 6.6 | |
| 6.5.4 | 5.5.5.4 | F.3.4 |
| 6.6 | 5.5.6 | |
| 6.6.2.1 | 5.5.6.1 | F.1.5 |
| 6.6.2.2 | 5.5.6.2 | |
| 6.6.2.3 | 5.5.6.3 | F.1.5/F.3.4 |
| 6.6.2.4 | 5.5.6.4 | |
| 6.6.2.5 | 5.5.6.5 | |
| 6.6.3 | 6.6 | |
| 7.1 | 5.5/6.2/6.8 | F.1.5 |
| 7.3 | 5.1.3 | |
| 7.5 | | F.3.1.6 |
| 8.1 | 5.5.1.1 | F.1.5 |
| 8.2.1 | 5.5.1.1 | F.3.4.2 |


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

| Clause in ISO/IEC/IEEE 14764 | Clause in ISO/IEC 12207 | Clause in ISO/IEC 12207, Amd 1 |
|---|---|---|
| 8.2.3 | 5.1/5.2 | |
| 8.2.4 | 5.5.1.1 | F.3.4.2 |
| 8.3.3 | 5.5.1.1 | F.3.4 |

--`,,```,,,,````-`-`,,`,,`,`,,`---

Copyright © 2006 IEEE. All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

# Annex B
(informative)

# Abbreviations

| | |
|---|---|
| CASE | computer-aided software engineering |
| CM | configuration management |
| IEC | International Electrotechnical Commission |
| IEEE | Institute of Electrical and Electronics Engineers |
| ISO | International Organization for Standardization |
| JTC | Joint Technical Committee |
| MR | modification request |
| MRs | modification requests |
| PR | problem report |
| PRs | problem reports |
| SEE | software engineering environment |
| STE | software test environment |


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

# Annex C
(informative)

# Bibliography

[1] IEEE 730-2002, IEEE Standard for Software Quality Assurance Plans.

[2] IEEE 828-1998, IEEE Standard for Software Configuration Management Plans.

[3] IEEE 829-1998, IEEE Standard for Software Test Documentation.

[4] IEEE 1012-1998, IEEE Standard for Software Verification and Validation.

[5] IEEE 1012a-1998, Supplement to IEEE Standard for Software Verification and Validation: Content Map to IEEE/EIA 12207.1-1997.

[6] IEEE 1028-1997(R2002), IEEE Standard for Software Reviews.

[7] IEEE 1042-1987, IEEE Guide to Software Configuration Management.

[8] IEEE 1058-1998, Standard for Software Project Management Plans.

[9] IEEE 1061-1998, IEEE Standard for a Software Quality Metrics Methodology.

[10] IEEE 1219-1998, IEEE Standard for Software Maintenance.

[11] IEEE 1348-1995, IEEE Recommended Practice for the Adoption of Computer-Aided Software Engineering (CASE) Tools.

[12] ISO/AFNOR:1989, Dictionary of Computer Science.

[13] ISO 9001:2000, Quality management systems -- Requirements.

[14] ISO/IEC 9003:2004, Software engineering -- Guidelines for the application of ISO 9001:2000 to computer software.

[15] ISO/IEC 14102:1995, Information technology -- Guideline for the evaluation and selection of CASE tools.

[16] ISO/IEC 14143-1: 1998, Information technology – Software measurement – Functional size measurement – Part 1: Definition of concepts.

[17] ISO/IEC TR 14471:1999, Information technology -- Software engineering -- Guidelines for the adoption of CASE tools.

[18] ISO/IEC TR 15271:1998, Information technology -- Guide for ISO/IEC 12207 (Software Life Cycle Processes).

[19] ISO/IEC 15288:2002, Systems engineering -- System life cycle processes.

[20] ISO/IEC TR 15846:1998, Information technology -- Software life cycle processes--Configuration management.

Copyright © 2006 IEEE. All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

[21] ISO/IEC 19759-2004, Software Engineering -- Guide to the Software Engineering Body of Knowledge – SWEBOK.

[22] ISO/IEC 2382-20:1990, Information technology -- Vocabulary; Part 20: System development.


Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale
Copyright © 2006 IEEE. All rights reserved

Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale

**ICS 35.080**

Price based on 44 pages

© ISO/IEC 2006 – All rights reserved
Copyright International Organization for Standardization
Provided by IHS under license with ISO
No reproduction or networking permitted without license from IHS
Not for Resale