INTERNATIONAL STANDARD

ISO/IEC 14764

First edition
1999-11-15

Information technology — Software maintenance

Technologies de l'information — Maintenance du logiciel

Reference number
ISO/IEC 14764:1999(E)

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 copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 734 10 79
E-mail copyright@iso.ch
Web www.iso.ch

Printed in Switzerland

Contents

Figures

Foreword

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. 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.

International Standard ISO/IEC 14764 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* Subcommittee SC 7, *Software engineering*.

Annex A of this International Standard is for information only.

Introduction

This International Standard clarifies requirements for 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 - Part 1: 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 and provides guidance. This International Standard elaborates the Maintenance Process contained in ISO/IEC 12207. 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.

In many projects, especially those having a long life, software maintenance will almost certainly be an important project consideration.

Due to product cost and time-frame constraints, as well as not following the best practices of ISO/IEC 12207, software is often delivered in an imperfect state. It is then necessary to be able to correct faults that are found during operation. It is frequently necessary to make improvements to the software to meet changed user requirements. Software maintenance may be a significant portion of life cycle costs.

This International Standard addresses readers who are familiar with Software Maintenance. It is recommended that readers who are unfamiliar with Software Maintenance study textbooks or obtain training before applying this International Standard.

Software Maintenance may be performed by a combination of software 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 this is agreement and organizationally dependent. The Software Maintenance requirements remain the same irrespective of the tools by which Software Maintenance is implemented.

Clause 1 provides the scope. Clause 2 provides conformance information. Clause 3 provides normative references. Clause 4 provides terms and definintions. Clause 5 provides the application of this International Standard. Clause 6 provides implementation considerations for the maintenance process. Clause 7 provides the software maintenance stragegy. Clause 8 provides the details of the maintenance process. Annex A provides a cross reference between clauses in this International Standard and ISO/IEC 12207.

A major contributor to this International Standard was IEC/TC 56.

Information technology – Software maintenance

1 Scope

This International Standard describes in greater detail management of the Maintenance Process described in ISO/IEC 12207. 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 includes maintenance for multiple software products with the same maintenance resources. "Maintenance" in this International Standard means software maintenance unless otherwise stated.

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

1.1 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.

1.2 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.

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 users of off-the-shelf software products unless incorporated into a deliverable product (ISO/IEC 12207). For example, organizations may wish to use this International Standard when maintaining word processing templates or macros used across the organization.

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) or the approach used in development (e.g., Rapid Application, prototype, mock-up).

1.3 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.

The steps to apply this International Standard are contained in ISO/IEC TR 15271.

2 Conformance

Compliance with this International Standard is satisfied by complying with the requirements of ISO/IEC 12207.

3 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this International Standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this International Standard 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 2382-20, *Information technology — Vocabulary — Part 20: System development.*

ISO 5807, *Information processing — Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts.*

ISO 8402, *Quality management and quality assurance — Vocabulary.*

ISO/IEC 9126, *Information technology — Software product evaluation — Quality characteristics and guidelines for their use.*

ISO/IEC 12207, *Information technology — Software life cycle processes.*

4 Terms and definitions

For the purposes of this International Standard, the terms and definitions given in ISO/IEC 12207, ISO 8402, ISO/IEC 2382-1 and ISO/IEC 2382-20, and the following apply.

4.1

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.

4.2

baseline

a formally approved version of a Configuration Item, regardless of media, formally designated and fixed at a specific time during the Configuration Item's life cycle [ISO/IEC 12207]

NOTE Sometimes a new baseline is referred to as a new release.

4.3

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.

4.4

Maintainability Plan

a document setting out the specific maintainibility practices, resources and sequence of activities relevant to software

NOTE The developer prepares the Maintainibility Plan.

4.5

maintenance enhancement

a maintenance enhancement is a software change which is not a software correction

NOTE There are two types of software enhancements, adaptive and perfective.

4.6

Maintenance Plan

a document setting out the specific maintenance practices, resources, and sequence of activities relevant to maintaining a software product

NOTE The maintainer prepares the Maintenance Plan. The Plan should be activated once a product transitions to the maintenance phase.

4.7

Maintenance Process

the Maintenance Process contains the activities and tasks of the maintainer

NOTE This process is activated when the software product undergoes modifications to code and associated documentation due to a problem or the need for improvement adaptation. The objective is to modify existing software product preserving its integrity. This process includes the migration and retirement of the software product.

4.8

maintenance programme

the organizational structure, responsibilities, procedures, processes, and resources used for implementing the Maintenance Plan

NOTE The term “programme” is synonymous with “infrastructure.”

4.9

Modification Request (MR)

a generic term used to identify proposed changes 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 type maintenance. MRs are also referred to as change requests.

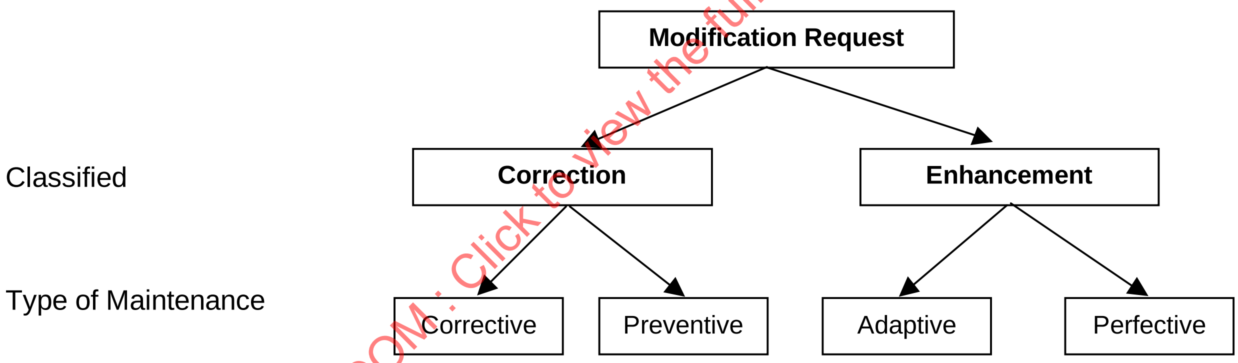

Figure 1 — Modification Request

4.10

perfective maintenance

the modification of a software product after delivery to improve performance or maintainability

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

4.11

preventive maintenance

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

4.12

Problem Report (PR)

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

4.13

Software Engineering Environment (SEE)

the set of automated tools, firmware devices, and hardware necessary to perform the software engineering effort

NOTE The automated tools may include but are not limited to compilers, assemblers, linkers, loaders operating systems, debuggers, simulators, emulators, test tools, documentation tools, and database management systems.

4.14

Software Test Environment (STE)

the facilities, hardware, software, firmware, procedures, and documentation needed to perform qualification, and possibly other, testing of software

NOTE Elements may include but are not limited to simulators, code analyzers, test case generators, and path analyzers, and may also include elements used in the software engineering environment. [MIL-HDBK-347]

4.15

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.

5 Application of this International Standard

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

5.1 Maintenance process

Software 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 Software Maintenance primary life cycle process through an agreement or contract. The Operation primary life cycle process of ISO/IEC 12207 may initiate the Software Maintenance life cycle process through submission of a Modification Request or Problem Report. The Software Maintenance primary 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 Software Maintenance life cycle process.

The organizational life cycle processes of ISO/IEC 12207 consist of four processes. The Management, Infrastructure, and Training organizational life cycle processes of ISO/IEC 12207 are employed by the maintainer when initiating each maintenance project. The Improvement Process of ISO/IEC 12207 is invoked to effect software maintenance process improvement.

Tailoring of this International Standard is addressed in ISO/IEC 12207. Tailoring is appropriate for non-routine events such as emergency maintenance.

5.2 Organization of this International Standard

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

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

6 Implementation 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 implementation considerations.

The Maintenance Process is needed because software products undergo change over the life cycle. 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.

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 Modification Requests (MRs)/Problem Reports (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 Maintenance Process.

Maintenance may also be required regardless of the development life cycle model (e.g., incremental, waterfall, evolutionary) or the approach used in development (e.g., Rapid Application, prototype, mock-up). For example, upon release of a new build increment, development continues but maintenance may be required for the installed increment.

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 changes necessitated by actual errors in a software product. If the software product does not meet its requirements, corrective maintenance is performed

Preventive Maintenance refers to the changes necessitated by detecting potential errors in a software product. Preventive Maintenance is commonly performed on software products which have safety or prevention of loss of life as a concern.

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

Software maintenance requires change 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 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.

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. 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 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 effects on software operation (e.g., emergency releases, periodic releases);
- ways in which the acquirer will be informed on the status of current or future changes;
- methods to confirm that the changes will not introduce other problems into the software;
- classification of type of change, urgency, and relationship with other pending proposed changes.

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 DTR 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 DTR 14471].

6.5 Software 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. A process should be implemented to identify, define, select, apply, validate, and improve software measurement for software maintenance.

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 metric data should be collected in order to assist life cycle cost estimating.

6.6 Documentation of process

The detailed software Maintenance Process (clause 8 of this International Standard) should be documented so that all maintenance personnel follow the same process. The metrics 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;

— ensure the supportability of the software product;

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

Planning is discussed in detail in clause 7 of this International Standard. Supportability of the software product includes tasks such as testing, and ensuring maintainability. ISO/IEC 9126 addresses maintainability and other characteristics that are important considerations during developments. Supportability 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.

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 should be used to define and assess the quality of software. Both qualitative and quantitative evaluation is important. Maintainability is a software quality characteristic that affects the speed and ease of changing software after it is released for use (ISO/IEC 9126).

6.8.1 Maintainability and the development process

Maintainability should be addressed prior to software development. Agreements should be developed between the acquirer and supplier as part of the Initiation Activity of ISO/IEC 12207. The developer should prepare a Maintainability Plan which provides specific maintainability practices, resources and sequence of activity relevant to software. An effort to support the monitoring and evaluation of the maintainability aspects of the Development Process of ISO/IEC 12207 should be established.

The capability to monitor and evaluate maintainability aspects 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 are used to define the techniques employed to facilitate maintenance costing and resources. Quantitative requirements are used to define maintainability magnitudes or quality criteria and the metrics used to determine values or indicators throughout the various software life cycle phases.

The effectiveness of such an effort during development becomes evident once maintenance activities begin. Developers should implement requirements for maintainability and maintainers should monitor implementation. 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 (Guide for ISO/IEC TR 15271). Accordingly, a maintenance strategy should be developed and maintenance should be planned (clause 7 of this International Standard).

A software maintenance strategy should also be established prior to design. This early maintainer involvement in the development has the potential to save money. There are many actions, including software maintenance planning, to be performed during the development process. These actions should be documented in the software maintenance plan (sub-clause 7.3.2 of this International Standard).

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;
- 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;
- the life of the various development tools.

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;

— 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). The main features of this Development Process Activity of ISO/IEC 12207 which effect 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 proved their dependability.

Modular design, combined with top-down analysis, and adequate documentation, which allow for easy additions when necessary, are the two main features which will continue to achieving maintainability requirements.

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 accurate, detailed description of each function to complete the proposed programming solution. Software maintainability will be improved by inclusion of quality characteristics found in ISO/IEC 9126.

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 with quality documentation include:

— ensuring legibility;

— avoiding non-structured code;

— eliminating classic traps by considering the language weaknesses;

— detecting errors in the detailed design;

— using techniques to facilitate error-tracing.

6.8.2.5 Software qualification testing

This activity ensures that the implementation of each software requirement is tested for compliance (ISO/IEC 12207). The software requirements which relate to quality are tested during this activity. 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 avoid repeating the same errors 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, 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).

6.10 Documentation

Maintainers are often faced with providing maintenance for a software product for which little or no documentation exists. If no documentation exists, the maintainer should create the needed documentation. Documentation creation is a part of perfective maintenance. This presents difficulty in performing the maintenance function. When faced with this situation, maintainers should perform the following 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 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) Fix low priority MRs or PRs.

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. 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.

7.2 The maintenance concept

Determining the Maintenance Concept should be the first step in developing the software maintenance strategy. 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;
- the tailoring of the 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 Tailoring of the process

The Maintenance Concept should address the tasks of post-delivery software maintenance. Different organizations may perform different tasks during maintenance. An early attempt should be made to identify these organizations and to document them in the Maintenance Concept. The Maintenance Concept should also reflect the Maintenance Process that will be employed.

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:

- life of the software product;
- long term costs;

- 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 SEE and STE;
- personnel costs such as salaries and benefits.

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

7.3 Maintenance planning

7.3.1 Introduction

Maintenance Planning's purpose 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.

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 changes 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;
- what resources will be available for maintenance;
- where maintenance will be performed;

— when maintenance will commence.

7.3.3 Guidelines for a maintenance plan

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.

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) Describe any agreement protocols between customer and supplier

b) Maintenance concept

   1) Describe the concept

   2) Describe the level of support for the system

   3) Identify the support period

   4) Tailor the process

c) Organization and maintenance activities

   1) Pre-Delivery roles and responsibilities of the maintainer

      i) Process Implementation

      ii) Establish Infrastructure

      iii) Establish Training Process

      iv) Establish the Maintenance Process

   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)

      viii) Train Personnel (Maintainer and User), as applicable

ix) Improve the Process

3) Role of the user

i) Acceptance Testing

ii) Interface with other organizations

d) 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) Documentation

i) Software Quality Plan

ii) Project Management Plan

iii) Configuration Management Plan

iv) Development documents

v) Maintenance manuals

vi) Verification Plan

vii) Validation Plan

viii) Test Plan, Test Procedures, and Test Reports

ix) Training Plan

x) User's Manual(s)

6) Data

7) Other resource requirements (if needed)

e) 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) Tailored process

f) Training

1) Identify training needs of the Maintainer and Users

g) Maintenance records and reports

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

2) Status of requests by categories

3) Priorities of requests

4) Metric 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.

7.4.1 Personnel resources

One of the major issues in planning for software maintenance is the issue of software maintenance resource requirements planning. Personnel requirements are a major cost factor, and, at the same time, the most difficult to accurately estimate. The two most popular approaches for estimating resources for software maintenance are use of parametric models and use of experience.

Models require historical empirical data. The best approach in using experience is to have empirical, historical data.

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 Engineeeing Environments (SEEs) and Software Test Environment (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 effort 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.

8 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 is in existence and is functional prior to the 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 oututs. Outputs are the data or objects produced by the maintenance activity. Support identifies supporting and organizational life cycle processes of ISO/IEC 12207 used by the maintenance activities.

Fiigure 2 provides an overview of the Maintenance Process.

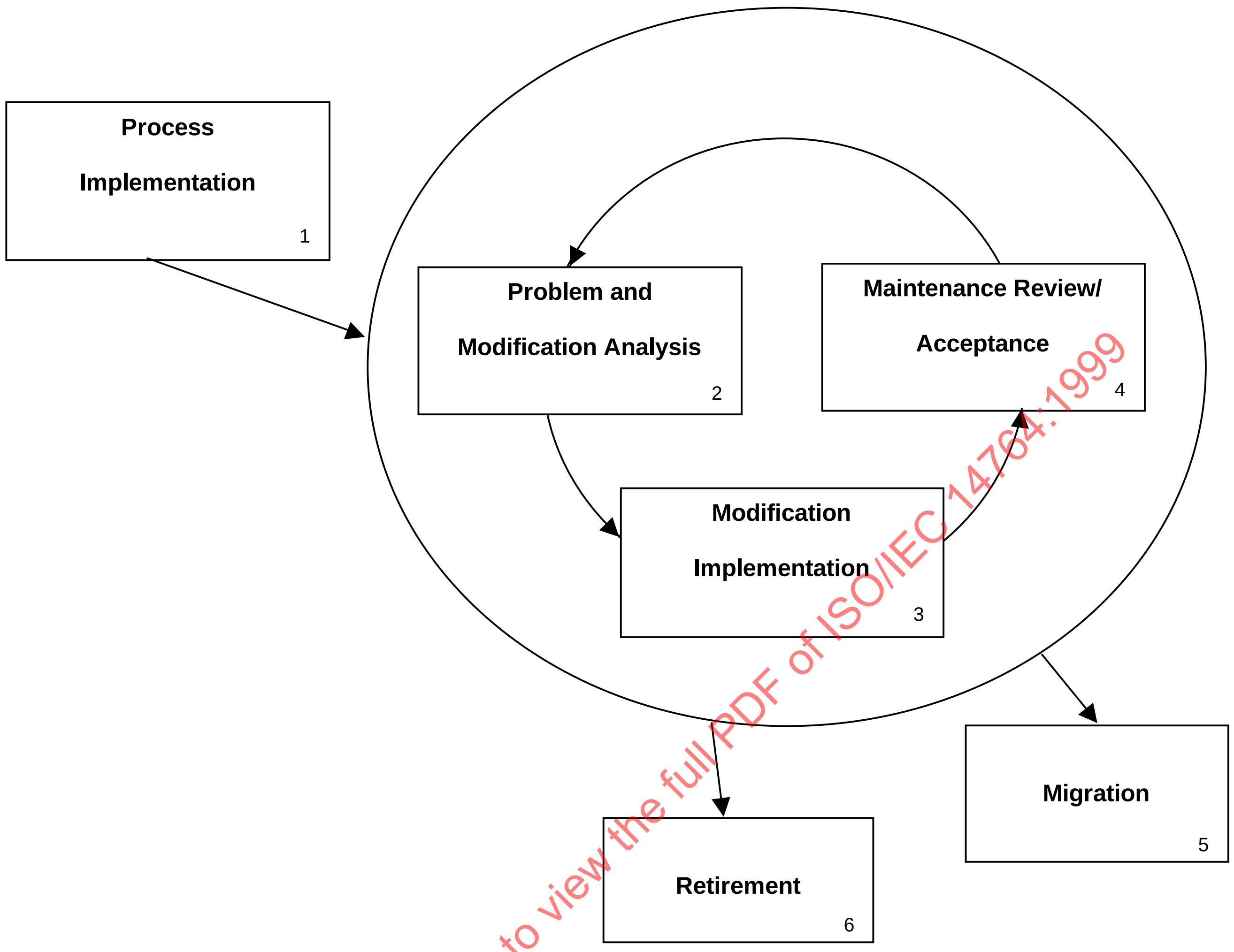

Figure 2 — Maintenance Process

8.1 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 7.3.2 of this International Standard) should be developed in parallel with the Development Plan. The maintainer should also establish needed organizational interfaces during this activity.

8.1.1 Inputs

The inputs for the Process Implementation activity should include:

- The Old Baseline;
- System Documentation;
- A Modification Request (MR) or Problem Report (PR).

8.1.2 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.

8.1.2.1 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.

8.1.2.2 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 be recorded and entered into the Problem Resolution Process (ISO/IEC 122007 clause 6.8).

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.

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 users.

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

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

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

8.1.2.3 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 clause 6.2) for managing modifications to the existing system.

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

8.1.3 Controls

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

8.1.4 Support

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

- Documentation Process;
- Configuration Management Process;
- Quality Assurance Process;
- Joint Review Process;
- Management Process;
- Infrastructure Process;
- Training Process.

8.1.5 Outputs

The outputs of this activity are:

- The Maintenance Plan;
- Maintenance Procedures;
- Problem Resolution Procedures;
- Plans for User Feedback;
- The Transition Plan;
- Configuration Management Plan.

All outputs should be placed under configuration management.

8.2 Problem and modification analysis

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 implementation 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.

8.2.1 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.

8.2.2 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.

8.2.2.1 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:

d) Determine if the maintainer is adequately staffed to implement the proposed change.

e) Determine if the program is adequately budgeted to implement the proposed change.

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

g) Determine the operational issues to be considered. For example, what are the anticipated changes 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).

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

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

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

k) Determine the impact on existing schedules.

l) Determine the level of test and evaluation required.

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

8.2.2.2 Verification

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

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

8.2.2.3 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 of the size and magnitude of the modification.

e) Develop at least three options to implement the modification.

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

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

8.2.2.4 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.

d) Update the status accounting database.

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

8.2.2.5 Approval

Before the system can be modified, 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 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).

8.2.3 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.

8.2.4 Support

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

— Documentation Process;

— Quality Assurance Process;

- Problem Reporting Process.

8.2.5 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;
- 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 Development Folders;
- Updated Requirements.

8.3 Modification implementation

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

8.3.1 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.

8.3.2 Tasks

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

8.3.2.1 Analysis

Once the MR/PR has been approved, the maintainer shall (ISO/IEC 12207 sub-clause 5.5.3.1) conduct analysis 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 development folders (SDFs). 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 SDFs.

8.3.2.2 Development process

The maintainer shall (ISO/IEC 12207 sub-clause 5.5.3.2) enter the Development Process (ISO/IEC 12207 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 Development Process should be tailored to meet the needs of the modification effort.

8.3.3 Controls

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

8.3.4 Support

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

— Documentation Process;

— Quality Assurance Process;

- Joint Review Process.

8.3.5 Outputs

The outputs of this activity should include:

- Updated Test Plans and Procedures;
- Updated Documentation;
- Modified Source Code;
- Test Reporting;
- Metrics.

The updated documentation should include:

- Updated Modification Records;
- Detailed Analysis Report;
- Updated Requirements;
- Updated Test Plans, Test Procedures, and Test Reports;
- Updated Training Materials.

8.4 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.

8.4.1 Inputs

The inputs to the Maintenance Reviews/Acceptance activity are:

- The Modified Software;
- Modification Test Results.

8.4.2 Tasks

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

8.4.2.1 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.

c) Verify that coding standards were complied with.

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) Perform testing.

h) Develop test report.

8.4.2.2 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) Conduct functional and physical configuration audits.

8.4.3 Controls

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

8.4.4 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.

8.4.5 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.

8.5 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.

8.5.1 Inputs

The inputs to the Migration activity are:

- The Old Environment;
- The New Envrionment;
- The Old Baseline;
- The New Baseline.

8.5.2 Tasks

The Maintainer effects the migration by complying 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.

8.5.2.1 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 is 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.

8.5.2.2 Migration plan

In order to adequately control the migration of a system, 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:

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.

8.5.2.3 Notification of intent

Once the maintainer has completed the planning of the migration, 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.

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.

8.5.2.4 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.

8.5.2.5 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.

d) Remove the old equipment.

8.5.2.6 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.