

Recognized as an
American National Standard (ANSI)

IEEE Std 1062, 1998 Edition (R2002)
(Includes IEEE Std 1062-1993
and IEEE Std 1062a-1998)

IEEE Recommended Practice for Software Acquisition

Sponsor

Software Engineering Standards Committee
of the
IEEE Computer Society

Approved 21 January 2003
American National Standards Institute

IEEE Std 1062-1993 Reaffirmed 11 September 2002
IEEE Std 1062-1993 Approved 2 December 1993
IEEE Std 1062a-1998 Approved 8 December 1998
by the

IEEE-SA Standards Board

Abstract: A set of useful quality practices that can be selected and applied during one or more steps in a software acquisition process is described. This recommended practice can be applied to software that runs on any computer system regardless of the size, complexity, or criticality of the software, but is more suited for use on modified-off-the-shelf software and fully developed software.
Keywords: acquirer, modified-off-the-shelf software, software acquisition life cycle, software acquisition process, supplier

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 1998 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 22 December 1998. Printed in the United States of America.

Print: ISBN 0-7381-0334-9 SH94656
PDF: ISBN 0-7381-1514-2 SS94656

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

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

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

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

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

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

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

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

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

Introduction

(This introduction is not a part of IEEE Std 1062, 1998 Edition, IEEE Recommended Practice for Software Acquisition.)

This introduction provides some background on the rationale used to develop this recommended practice. This information is meant to aid in the understanding and usage of this recommended practice.

This recommended practice describes the management and execution of software acquisition activities. It is intended for:

- Individuals or organizations that use software and acquire that software from suppliers;
- Individuals or organizations that acquire software from a developer for resale to other individuals or organizations;
- Individuals or organizations that influence how software is acquired from suppliers;
- Suppliers interested in providing high-quality software to acquirers.

This recommended practice is designed to help organizations and individuals:

- Incorporate quality considerations during the definition, evaluation, selection, and acceptance of supplier software for operational use;
- Determine how supplier software should be evaluated, tested, and accepted for delivery to end users.

This recommended practice is intended to satisfy the following objectives:

- Promote consistency within organizations in acquiring third-party software from software suppliers;
- Provide useful practices on including quality considerations during acquisition planning;
- Provide useful practices on evaluating and qualifying supplier capabilities to meet user requirements;
- Provide useful practices on evaluating and qualifying supplier software;
- Assist individuals or organizations judging the quality of supplier software for referral to end users.

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

Participants

IEEE Std 1062-1993 was prepared by the Software Acquisition Working Group. At the time it was approved, the working group consisted of the following members:

Phillip C. Marriott, *Chair*

Karen Allison
Ellen M. Augustine
Mordechai Ben-Menachem
David L. Boudreau
Rick Burgess
Edward R. Byrne
Liliane Choney
Christopher Cooke
Rena Crabill
Patricia M. Daggett
Linda Uca Domiani
Dennis J. Eaton
Leo G. Egan
Jane Frederick
Thomas Gray

Flo Harteloo, *Vice Chair*

Nicholsa P. Ginex
Paul Haller
Frank Henninger
Lloyd Johnson
Anna Johnston
Miles Kehoe
Bob Kessler
C. L. (Kelly) Kjelstrom
Larry King
Thomas M. Kurihara
Rebecca Ann Lamb
A. Lip Lim
F. C. Lim
Ben Livson
John J. McKissick, Jr.

Louis R. Mills
Rafael E. Padilla
Duane K. Palmiter
John Passafiume
Sandy Raddue
Robert H. Randolph
Horst P. Richter
David P. Schwartz
Mark Shelton
David M. Siefert
Paulette Spink
Kenneth Thoreson
William Timby
George W. Trever
William S. Turner III

The following persons balloted IEEE Std 1062-1993:

W. Arfvidson	Y. Gershkovitch	P. C. Marriott
M. Azuma	J. Gonzalez-Sanz	R. Martin
B. Banerjee	D. A. Gustafson	I. Mazza
M. Ben-Menachem	W. Hefley	D. E. Nickle
H. R. Berlack	C. P. Hollocker	J. G. Phippen
W. J. Boll, Jr.	J. W. Horch	J. D. Pope
K. L. Briggs	D. Johnson	S. R. Schach
B. Brocka	P. Klopfenstein	H. Schaefer
F. Buckley	T. M. Kurihara	W. A. Schnoege
F. Coallier	R. A. Lamb	G. D. Schumacher
P. W. Daggett	J. B. Lane	D. M. Siefert
T. Daughtrey	F. C. Lim	L. L. Tripp
L. G. Egan	B. Lindberg	D. Wallace
C. L. Evans	B. Livson	A. Wilson
J. W. Fendrich	J. Maayan	P. A. Wolfgang
R. G. Fordham	H. Mains	N. C. Yopconka

IEEE Std 1062a-1998 was prepared by the Life Cycle Data Harmonization Working Group of the Software Engineering Standards Committee of the IEEE Computer Society. At the time it was approved, the working group consisted of the following members:

Leonard L. Tripp, Chair

Edward Byrne	Dennis Lawrence	Terry Rout
Paul R. Croll	David Maibor	Richard Schmidt
Perry DeWeese	Ray Milovanovic	Norman F. Schneidewind
Robin Fralick	James Moore	David Schultz
Marilyn Ginsberg-Finner	Timothy Niesen	Basil Sherlund
John Harauz	Dennis Rilling	Peter Voldner
Mark Henley		Ronald Wade

The following persons balloted IEEE Std 1062a-1998:

Syed Ali	Julio Gonzalez-Sanz	James W. Moore
H. Ronald Berlack	L. M. Gunther	Pavol Navrat
Richard E. Biehl	David A. Gustafson	Myrna L. Olson
Juris Borzovs	Jon D. Hagar	Alex Polack
M. Scott Buck	John Harauz	Peter T. Poon
James E. Cardow	William Hefley	Kenneth R. Ptack
Keith Chan	Debra Herrmann	Ann E. Reedy
Antonio M. Cicu	John W. Horch	Annette D. Reilly
Theo Clarke	Jerry Huller	Dennis Rilling
Virgil Lee Cooper	Peter L. Hung	Terence P. Rout
W. W. Geoff Cozens	George Jackelen	Andrew P. Sage
Paul R. Croll	Frank V. Jorgensen	Stephen R. Schach
Gregory T. Daich	Vladan V. Jovanovic	Hans Schaefer
Geoffrey Darnton	William S. Junk	Norman Schneidewind
Taz Daughtrey	Ron S. Kenett	David J. Schultz
Bostjan K. Derganc	Judith S. Kerner	Robert W. Shillato
Perry R. DeWeese	Robert J. Kierzyk	David M. Siefert
Audrey Dorofee	Thomas M. Kurihara	Lynn J. Simms
Evelyn S. Dow	John B. Lane	Carl A. Singer
Sherman Eagles	J. Dennis Lawrence	Luca Spotorno
Leo Egan	Fang Ching Lim	Fred J. Strauss
Richard E. Fairley	John Lord	Sandra Swearingen
John W. Fendrich	Stan Magee	Toru Takeshita
Jay Forster	Harold Mains	Richard H. Thayer
Kirby Fortenberry	Robert A. Martin	Booker Thomas
Barry L. Garner	Patrick D. McCray	Leonard L. Tripp
Marilyn Ginsberg-Finner	Bret Michael	Theodore J. Urbanowicz
John Garth Glynn	Alan Miller	Glenn D. Venables

Ronald L. Wade
Dolores Wallace
William M. Walsh

John W. Walz
Scott A. Whitmire
P.A. Wolfgang

Paul R. Work
Geraldine Zimmerman
Peter F. Zoll

When the IEEE-SA Standards Board approved 1062a-1998 on 8 December 1998, it had the following membership:

Richard J. Holleman, *Chair*

Donald N. Heirman, *Vice Chair*

Judith Gorman, *Secretary*

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

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

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

*Member Emeritus

Valerie E. Zelenty
IEEE Standards Project Editor

Contents

1.	Overview.....	1
1.1	Scope.....	1
1.2	Terminology.....	2
2.	References.....	2
3.	Definitions.....	3
4.	Introducing the software acquisition process.....	4
4.1	Software acquisition life cycle.....	4
4.2	Nine steps in acquiring quality software.....	5
5.	Software acquisition process.....	8
5.1	Planning organizational strategy.....	8
5.2	Implementing organization's process	10
5.3	Defining the software requirements.....	11
5.4	Identifying potential suppliers	12
5.5	Preparing contract requirements	13
5.6	Evaluating proposals and selecting supplier.....	14
5.7	Managing for supplier performance.....	16
5.8	Accepting the software	17
5.9	Using the software	18
6.	Summary.....	18
Annex A	(informative) Checklists to assist organizations in establishing their own software acquisition process	19
Annex B	(normative) Acquisition Plan guidelines	35
Annex C	(informative) Guidelines for compliance with IEEE/EIA 12207.1-1997	39

IEEE Recommended Practice for Software Acquisition

1. Overview

This recommended practice is divided into six clauses. Clause 1 provides the scope of this recommended practice. Clause 2 lists references to other standards that are useful in applying this recommended practice. Clause 3 provides definitions that are either not found in other standards, or have been modified for use with this recommended practice. Clause 4 establishes the nine steps involved in a software acquisition process, relates each of these steps to a major acquisition phase, and identifies the key inputs and outputs of each step. Clause 5 describes the nine steps in a software acquisition process and the related quality practices that apply to acquiring software. In order to be in compliance with this recommended practice, an implementation must adhere to Clause 5. Clause 6 summarizes the successful way to acquire high-quality products and services from software suppliers.

This recommended practice also contains three annexes. Annex A provides a set of checklists that individuals or organizations may elect to adapt to their specific needs, Annex B provides acquisition plan guidelines, and Annex C provides guidelines for compliance with IEEE/EIA 12207.1-1997.

1.1 Scope

This is a recommended practice for performing software acquisitions. It describes a set of useful quality practices that can be selected and applied during one or more steps in a software acquisition process.

In this recommended practice, software products have been classified according to the degree to which the acquirer may specify the features of the software. They are: commercial-of-the-shelf (COTS), modified-off-the-shelf (MOTS), and fully developed item.

COTS software is stable and is normally well-defined in terms of documentation and known capabilities and limitations. It usually comes with “how to operate” documentation. COTS software is defined by a market-driven need. It is commercially available and its fitness for use has been demonstrated by a broad spectrum of commercial users. Also, the COTS software supplier does not advertise any willingness to modify the software for a specific customer.

MOTS software is similar to COTS software; however, MOTS software does advertise services to tailor the software to acquirer-specific requirements.

Fully developed software will often be unique for a specific application and will be produced on a one-of-a-kind or low-volume basis. The software typically will have the potential for future modification by the acquirer to meet changing needs. As a result, most of the documentation will be special to the project (with the exception of the supplier's standard documentation for the operating system, any standard application packages, and programming languages).

This recommended practice can be applied to software that runs on any computer system regardless of the size, complexity, or criticality of the software. However, this recommended practice is more suited for use on MOTS software and fully developed software. Each organization using this recommended practice will need to identify the classes of software to which this recommended practice applies and the specific quality characteristics and activities that need to be included within the acquisition process.

1.2 Terminology

The words *shall* and *must* identify the mandatory (essential) material within this recommended practice. The words *should* and *may* identify optional (conditional) material. The terminology in this recommended practice is based on IEEE Std 610.12-1990. New terms and modified definitions as applied in this recommended practice can be found in Clause 3.

2. References

The following standards are directly referenced in this recommended practice. Table 1 provides a cross-reference of standards that address topics related to software acquisition. These standards are binding to the extent referenced within the text of this recommended practice and are referenced to avoid duplication of requirements.

IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.¹

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

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

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.

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

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

IEEE Std 1016-1998, IEEE Recommended Practice for Software Design Descriptions.

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

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

IEEE Std 1045-1992, IEEE Standard for Software Productivity Metrics.

¹IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (<http://www.standards.ieee.org/>).

IEEE Std 1058-1998, IEEE Standard for Software Project Management Plans.

IEEE Std 1061-1998, IEEE Standard for Software Quality Metrics Methodology.

IEEE Std 1063-1987 (Reaff 1993), IEEE Standard for Software User Documentation.

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

IEEE Std 1209-1992, IEEE Recommend Practice for the Evaluation and Selection of CASE Tools.

IEEE Std 1219-1998, IEEE Standard for Software Maintenance.

ISO/IEC 9126: 1991, Information technology—Software product evaluation—Quality characteristics and guidelines for their use.²

Table 1—The relationship of software engineering standards to this recommended practice

Step	Standard	Role
All	IEEE Std 610.12-1990	Terminology
1	IEEE Std 1042-1987 IEEE Std 1058-1998 IEEE Std 730-1998	Description of quality assurance process Description of configuration management process Description of project management process
3	IEEE Std 830-1998 ISO/IEC 9126: 1991	Format and content of software requirements specification Definition of quality characteristics
4	IEEE Std 1209-1992	Description of software evaluation process
5	IEEE Std 730-1998 IEEE Std 829-1998 IEEE Std 1012-1998 IEEE Std 1016-1998 IEEE Std 1028-1997 IEEE Std 1061-1998 IEEE Std 1063-1987 IEEE Std 1074-1997	Description of quality assurance process Content of software test documentation Format and content of software V&V plan Content of design description Description of review and audit processes Description of software quality metric methodology Format and content of user documentation Description of developing software life cycle processes
7	IEEE Std 1045-1992 IEEE Std 1058-1998	Description of software productivity metrics Description of project management process
9	IEEE Std 1219-1998	Description of software maintenance process

3. Definitions

The definitions listed below establish meaning in the context of this recommended practice. Other definitions can be found in IEEE Std 610.12-1990. Note that for the purpose of this recommended practice, software includes documentation.

3.1 acquirer: A person or organization that acquires or procures a system or software product (which may be part of a system) from a supplier.

²ISO publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembe, CH-1211, Genève 20, Switzerland/Suisse (<http://www.iso.ch/>). ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA (<http://www.ansi.org/>).

3.2 acquisition: The process of obtaining a system or software product.

3.3 commercial-off-the-shelf (COTS): Software defined by a market-driven need, commercially available, and whose fitness for use has been demonstrated by a broad spectrum of commercial users.

3.4 contract: A binding agreement between two parties, especially enforceable by law or similar internal agreement wholly within an organization, for supply of service or for the supply, development, production, operation, or maintenance of a software product.

3.5 developer: A person or organization that performs development activities (including requirements analysis, design, testing through acceptance) during the software life cycle process.

3.6 modified-off-the-shelf (MOTS): Software product that is already developed and available, usable either “as is” or with modification, and provided by the supplier, acquirer, or a third party.

3.7 request for proposal (RFP): A document used by the acquirer as a means to announce intention to potential bidders to acquire a specified system or software product (which may be part of a system).

3.8 software acquisition process: The period of time that begins with the decision to acquire a software product and ends when the product is no longer available for use. The software acquisition process typically includes nine steps associated with planning the organizational strategy, implementing an organization’s process, determining the software requirements, identifying potential suppliers, preparing contract requirements, evaluating proposals and selecting the supplier, managing supplier performance, accepting the software, and using the software.

3.9 software: Computer programs, procedures, and associated documentation and data pertaining to the operation of a computer system (see also IEEE Std 610.12-1990).

3.10 software product: The complete set of computer programs, procedures, and associated documentation and data designated for delivery to a user.

3.11 statement of work: A document used by the acquirer as a means to identify, describe, and specify the tasks to be performed under the contract.

3.12 supplier: A person or organization that enters into a contract with the acquirer for the supply of a software product (which may be part of a system) under the terms of the contract.

4. Introducing the software acquisition process

4.1 Software acquisition life cycle

The software acquisition life cycle represents the period of time that begins with the decision to acquire a software product and ends when the product is no longer available for use. It typically includes a planning phase, contracting phase, product implementation phase, product acceptance phase, and follow-on phase. This life cycle provides an overall framework within which most software acquisitions occur.

The phases in the life cycle are broadly defined by a set of milestones that establish the beginning and ending of each phase. Some phases may have a longer duration than others, or may include more activities than other phases. These phases and their key milestones are

- a) *Planning phase.* This phase begins when the idea or need is established for acquiring software and ends when the request for proposal (RFP) is released.

- b) *Contracting phase.* After the RFP is released, this phase includes activities necessary to ensure that the supplier's products and services can satisfy the acquirer's quality criteria before signing the contract.
- c) *Product implementation phase.* This phase covers the period from contract signing until the software product has been received. A key activity is monitoring the supplier's efforts to ensure that all work and milestones are satisfactorily completed prior to delivery of the software product.
- d) *Product acceptance phase.* This phase includes all activities necessary to evaluate, test, and accept the software product. It begins when the software product is received and ends when the product is accepted.
- e) *Follow-on phase.* After the software product is accepted, this phase includes using the product to meet the acquirer's objectives and evaluating user satisfaction with the software product, its documentation, and support provided from the supplier. This phase continues until all provisions provided in the contract have been completed or until the software product is no longer available for use.

Each of these phases and their key milestones are summarized in Table 2. A special feature of this table includes a listing of the software acquisition process steps associated with each life cycle phase.

Table 2—Software acquisition phase milestones

Phase	Phase initiation milestone	Phase completion milestone	Steps in software acquisition process ^a
Planning	Idea is developed	Release the RFP	1) Planning organizational strategy 2) Implementing organization's process 3) Determining the software requirements
Contracting	RFP is released	Sign the contract	4) Identifying potential suppliers 5) Preparing contract requirements 6) Evaluating proposals and selecting the supplier
Product implementation	Contract is signed	Receive the software product	7) Managing supplier performance
Product acceptance	Software product is received	Accept the product	8) Accepting the software
Follow-on	Software product is accepted	Product is no longer in use	9) Using the software

^aThe step numbers referred to in this table correlate to the nine steps outlined in 4.2.

4.2 Nine steps in acquiring quality software

The software acquisition process provides a structure of major acquisition steps that are applicable to the acquisition of either fully developed software or MOTS software. The activities contained in each step all bear upon the development of a software product with the potential for high quality. Other related project objectives may also be improved upon or enhanced through the application of quality principles. Such objectives include on-time delivery and cost-effectiveness. Utilization of the activities in this process is expected to result in the delivery of high-quality, well-documented products.

A typical software acquisition process is provided in Figure 1 for understanding of the steps that an acquirer and supplier go through. Checklists have been provided in Annex A to assist in establishing a software acquisition process.

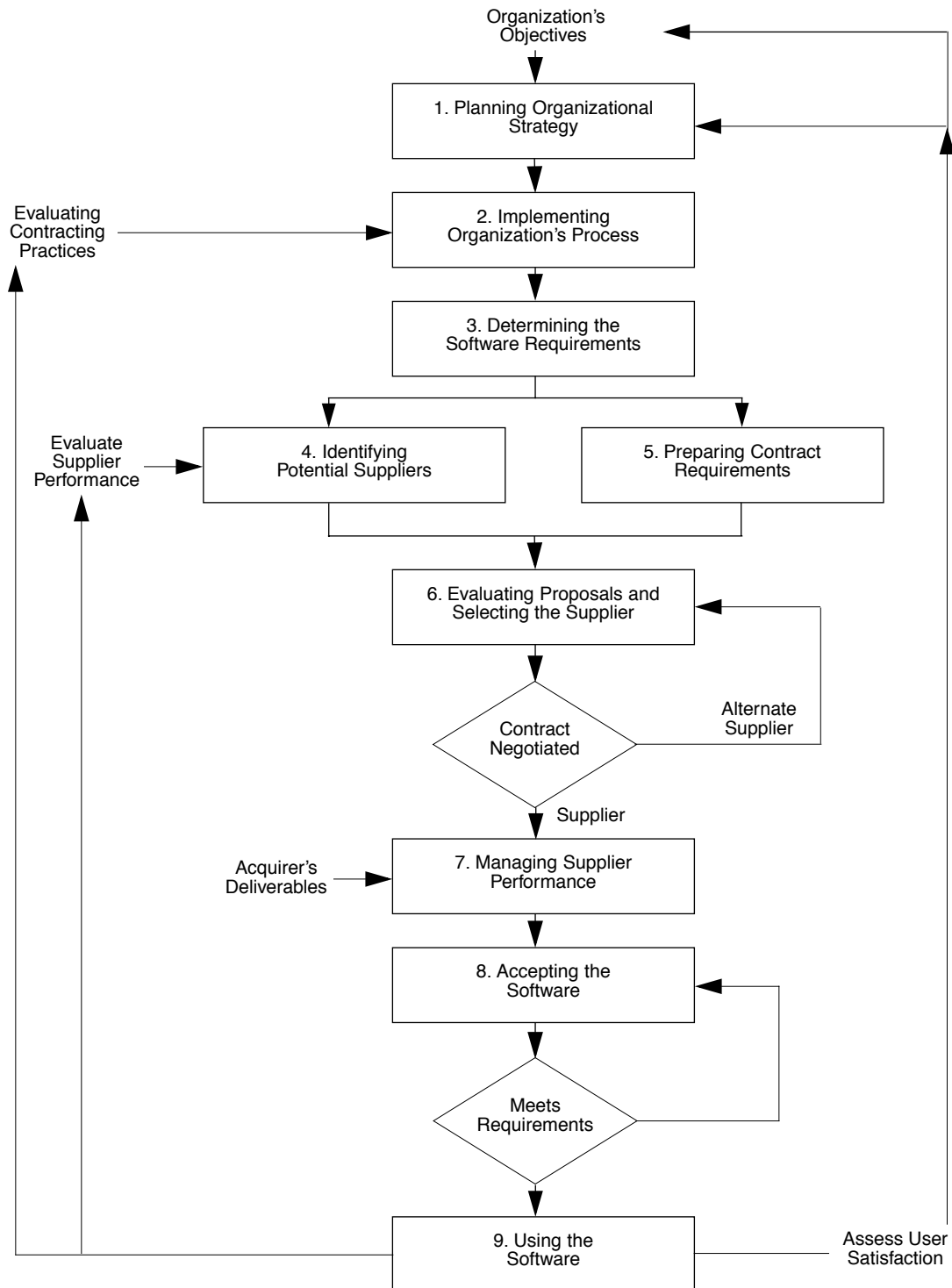
While this process may appear to define the acquisition of fully developed software, it can be tailored to fit the acquisition of MOTS software by selecting those activities relevant to that purpose.

The steps in this process include issues that organizations should consider when acquiring fully developed software. Some of the stated principles apply to contracting for any type of service and are covered in general acquisition guidance. However, these principles are included here for completeness and to emphasize that they can indeed be applied to fully developed software contracts as well as to others.

The software acquisition process is divided into nine steps. The steps are

- Step 1:* Planning organizational strategy. Review acquirer's objectives and develop a strategy for acquiring software.
- Step 2:* Implementing organization's process. Establish a software acquisition process that fits organization's needs for obtaining a quality software product. Include appropriate contracting practices.
- Step 3:* Determining the software requirements. Define the software being acquired and prepare quality and maintenance plans for accepting software supplied by the supplier.
- Step 4:* Identifying potential suppliers. Select potential candidates who will provide documentation for their software, demonstrate their software, and provide formal proposals. Failure to perform any of these actions is basis to reject a potential supplier. Review supplier performance data from previous contracts.
- Step 5:* Preparing contract requirements. Describe the quality of the work to be done in terms of acceptable performance and acceptance criteria, and prepare contract provisions that tie payments to deliverables. Review contract with legal counsel.
- Step 6:* Evaluating proposals and selecting the supplier. Evaluate supplier proposals, select a qualified supplier, and negotiate the contract. Negotiate with an alternate supplier, if necessary.
- Step 7:* Managing supplier performance. Monitor supplier's progress to ensure all milestones are met and to approve work segments. Provide all acquirer deliverables to the supplier when required.
- Step 8:* Accepting the software. Perform adequate testing and establish a process for certifying that all discrepancies have been corrected and that all acceptance criteria have been satisfied.
- Step 9:* Using the software. Conduct a follow-up analysis of the software acquisition contract to evaluate contracting practices, record lessons learned, and evaluate user satisfaction with the product. Retain supplier performance data.

These steps may overlap or occur in a different sequence, depending upon the organizational needs. One sequence is shown in Figure 1.

**Figure 1 — Software acquisition process**

Each step in the software acquisition process has certain key inputs and outputs. These are identified in Table 3. Note that outputs from various steps are also inputs to subsequent steps. Overall, defining the inputs and outputs provides a better understanding of each of the steps, which are explained in 5.1 through 5.9.

The subsequent clauses provide a solid basis for developing organizational procedures; they might also be helpful for developing ideas in an evaluation group.

5. Software acquisition process

5.1 Planning organizational strategy

When planning organization strategy do the following:

- a) Initiate a planning process;
- b) Set organizational strategy;
- c) Establish general practices.

5.1.1 Initiate a planning process

Initiate a planning process by

- a) Developing a scope for the planning process;
- b) Forming a planning group and reviewing the organization's objectives;
- c) Identifying the qualities a software product must possess to achieve the organization's objectives.

5.1.2 Set organizational strategy

Decide which quality characteristics the software should have as an aid in detailing a strategy for making acquisition of software. This strategy should include

- a) Developing a list of capabilities that would be helpful in identifying potential suppliers who could provide the needed software;
- b) Identifying responsibilities that are associated with either the supplier or the acquirer;
- c) Determining the extent of the supplier's organizational involvement in providing a high-quality product (consider the strategic areas shown in Annex A, checklist 1);
- d) Identifying those responsibilities that are best handled by the acquirer's organization or other internal sources;
- e) Identifying those responsibilities that are best included in a contract and negotiated with a supplier.

5.1.3 Establish general practices

Establish general practices to achieve consistency in negotiating and contracting with suppliers for software products. Practices for handling suppliers may be documented in a policy or operating procedure.

Table 3—Process steps—Key inputs and output

Phase	Steps ^a in software acquisition process	Inputs to the step ^b	Outputs from the step
<i>Planning</i>	1) Planning organizational strategy	- Acquirer's objectives - Strategic areas (1)	- Quality characteristics of software - Organizational strategy for acquiring software - General practices
	2) Implementing organization's process	- Steps 3–9 of the process (see 5.3) - Organizational strategy - Contracting practices - Organization's policies	- Establish a software acquisition process for organization - Supplier qualification and selection process
	3) Determining the software requirements	- Software definition (2 & 10) - Supplier evaluation criteria (3) - Acquirer and supplier obligations (4) - Quality plan and maintenance plan content (5)	- Software being acquired defined - Quality and maintenance plans defined - Proposal evaluation standards - Contingency plan - RFP
<i>Contracting</i>	4) Identifying potential suppliers	- Supplier performance data from prior contracts - Supplier evaluation criteria (3) - Definition of software - Results being acquired - User survey questionnaire (6)	- Information on software - MOTS software/suppliers - Candidate list - User survey
	5) Preparing contract requirements	- Supplier and acquirer responsibilities - Supplier performance standards (7) - Acquirer's terms and conditions - Quality assurance clauses - Payment provisions (8)	- Acceptance criteria - Supplier performance criteria - Evaluation and test criteria - Tie payments to deliverables - Prepared contract - Legal counsel review
	6) Evaluating proposals and selecting the supplier	- Supplier proposals - Proposal evaluation standards - Supplier qualification and selection process - Visit supplier facilities (3) - User survey results (6) - Quality assurance clauses	- Evaluation of proposals - Evaluation of suppliers - Qualified suppliers list - Supplier selection - Negotiated contract
<i>Product Implementation</i>	7) Managing supplier performance	- Negotiated contract - Contract milestones - Acquirer's deliverables provided to supplier - Monitor supplier progress (9) - Supplier performance criteria	- Work segments approved - Completed milestones - Software deliverables - Reliability and quality measurements - Feedback to supplier
<i>Product acceptance</i>	8) Accepting the software	- Acceptance criteria - Evaluation criteria (10) - Test criteria (11) - Quality plan - Maintenance plan - Supplier performance criteria - Establish acceptance process (12)	- Acceptance process - Acceptable software - Usable documentation
<i>Follow-on</i>	9) Using the software	- Software deliverables - Documentation - Support available - Quality plan - Maintenance plan	- Contracting practices evaluated - Practices to change - Practices to retain - User satisfaction assessment - Supplier performance data

^aThe step numbers referred to in this table correlate to the nine steps outlined in 4.2.^bThe numbers in parentheses refer to the checklists that are in Annex A.