

IEEE Recommended Practice for Software Acquisition

IEEE Computer Society

Sponsored by the
Software & Systems Engineering Standards (C/S2ESC) Committee



IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 1062™-2015
(Revision of
IEEE Std 1062-1993)

This is a Redline Document produced by Techstreet, a business of Thomson Reuters.

This document is intended to provide users with an indication of changes from one edition to the next. It includes a full-text version of the new document, plus an indication of changes from the previous version.

Redlines are designed to save time and improve efficiencies by using the latest software technology to find and highlight document changes. More professionals are using valuable new technologies like redlines, to help improve outcomes in a fastpaced global business world.

Because it may not be technically possible to capture all changes accurately, it is recommended that users consult previous editions as appropriate. In all cases, only the current base version of this publication is to be considered the official document.

Redline Processing Notes:

1. ~~Red Text~~ - Red strikethrough text denotes deletions.
2. Blue Text - Blue underlined text denotes modifications and additions.

IEEE Recommended Practice for Software Acquisition

Sponsor

Software & Systems Engineering Standards (C/S2ESC) Committee
of the
IEEE Computer Society

Approved 5 December 2015

IEEE-SA Standards Board

Abstract: A set of useful quality considerations that can be selected and applied during one or more steps in a software acquisition process is described in this recommended practice. The recommended practices can be applied to software that runs on any computer system regardless of the size, complexity, or criticality of the software. The software supply chain may include integration of off-the-shelf (OTS), custom, or free and open source software (FOSS). Each organization or individual using this recommended practice will need to identify the specific quality and activities that need to be included within its acquisition process.

Keywords: acquirer, custom developed, end user, FOSS, IEEE 1062™, off-the-shelf, OTS, SaaS, software acquisition process, software as a service, supplier

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

Copyright © 2016 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 26 February 2016. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-0085-5 STD20506
Print: ISBN 978-1-5044-0086-2 STDPD20506

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

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.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading “Important Notice” or “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

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

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

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.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive 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 comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA Website at <http://ieeexplore.ieee.org/xpl/standards.jsp> or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at <http://standards.ieee.org>.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: <http://standards.ieee.org/findstds/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at <http://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this IEEE recommended practice was completed, the RP of the Software Acquisition Working Group had the following membership:

Marcus Hervey, *Chair*
Paul Bishop, *Vice Chair*
Philip Bernick, *Secretary*

Angela Anuszewski
Taohong Chang
Milton Concepcion
Steve Cropper
Suellen Eslinger
Michael George
Guy Goodwin
Jay Griesser

Leslie Holloway
Thomas Howard
Kathleen Kaplan
Ronald Kohl
Dana Kusumo
Mario Lozano
Linda Martinez
Winifred Menezes

Zvikomborero Murahwi
Lori Naiberg-Olson
Annette Reilly
Alberto Sampaio
Rob Schaaf
Mike Smith
Joan Williamson
Sharon Young

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

Johann Amsenga
Bakul Banerjee
Pieter Botman
Susan Burgess
Juan Carreon
Sue Carroll
Suresh Channarasappa
Keith Chow
S Claassen
Raul Colcher
Paul Croll
Geoffrey Darnton
Ronald Dean
Sourav Dutta
Andrew Fieldsend
Eva Freund
David Fuschi
Gregg Giesler
Randall Groves
Louis Gullo
John Harauz

Mark Henley
Richard Hilliard
Werner Hoelzl
Bernard Homes
Noriyuki Ikeuchi
Atsushi Ito
Mark Jaeger
Cheryl Jones
Adri Jovin
Piotr Karocki
Yuri Khersonsky
Thomas Kurihara
Dewitt Latimer
Claire Lohr
Edward McCall
James W. Moore
Michael Newman
Olileanya Ogbonna
Robert Peterson
Annette Reilly
Robert Robinson
Randy Saunders

Bartien Sayogo
Robert Schaaf
Hans Schaefer
Maud Schlich
Carl Singer
Michael Smith
Kendall Southwick
Luca Spotorno
Thomas Starai
Eugene Stoudenmire
Walter Struppler
Marcy Stutzman
Michael Swearingen
Thomas Tullia
John Vergis
David Walden
Michael Waterman
Stephen Webb
Jian Yu
Oren Yuen
Daidi Zhong

When the IEEE-SA Standards Board approved this recommended practice on 5 December 2015, it had the following membership:

John D. Kulick, *Chair*
Jon Walter Rosdahl, *Vice Chair*
Richard H. Hulett, *Past Chair*
Konstantinos Karachalios, *Secretary*

Masayuki Ariyoshi
Ted Burse
Stephen Dukes
Jean-Philippe Faure
J. Travis Griffith
Gary Hoffman
Michael Janezic

Joseph L. Koepfinger*
David J. Law
Hung Ling
Andrew Myles
T. W. Olsen
Glenn Parsons
Ronald C. Petersen
Annette D. Reilly

Stephen J. Shellhammer
Adrian P. Stephens
Yatin Trivedi
Phillip Winston
Don Wright
Yu Yuan
Daidi Zhong

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 1062-2015, 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 the following:

- Individuals or organizations that acquire 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 software acquisition planning
- Provide useful practices on evaluating and qualifying supplier capabilities to meet user software 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

Contents

1. Overview	11
1.1 Scope	11
1.2 Purpose	12
2. Normative references.....	12
3. Definitions and terms	13
3.1 Definitions	13
3.2 Use of should, may, and can.....	13
4. Software acquisition alternatives.....	13
4.1 Introduction	13
4.2 Custom-developed software	14
4.3 Off-the-shelf (OTS) software.....	14
4.4 Software as a service (SaaS).....	15
5. Software acquisition process	18
5.1 Purpose	18
5.2 Eight steps in acquiring quality software.....	19
5.3 Step 1: Planning the software acquisition strategy	22
5.4 Step 2: Defining the acquisition and software requirements	24
5.5 Step 3: Identifying potential suppliers	26
5.6 Step 4: Preparing contract requirements.....	28
5.7 Step 5: Evaluating proposals and selecting the supplier	30
5.8 Step 6: Managing for supplier performance	32
5.9 Step 7: Accepting the software	33
5.10 Step 8: Evaluating the process and identifying improvement opportunities.....	34
6. Quality assurance for software acquisition	36
6.1 Objectives of quality assurance in software acquisition	36
6.2 Implementing quality assurance in software acquisition	36
Annex A (informative) Checklists for quality software acquisition processes.....	38
A.1 Checklist 1: Organizational strategy.....	39
A.2 Checklist 2: Define the software.....	40
A.3 Checklist 3: Supplier selection evaluation.....	42
A.4 Checklist 4: Supplier and acquirer obligations	44
A.5 Checklist 5: Quality and maintenance plans	45
A.6 Checklist 6: User survey	46
A.7 Checklist 7: Supplier performance standards.....	49
A.8 Checklist 8: Contract payments	50
A.9 Checklist 9: Monitor supplier progress.....	51
A.10 Checklist 10: Software evaluation	52
A.11 Checklist 11: Software testing	55
A.12 Checklist 12: Software acceptance	56
Annex B (informative) Software safety assurance and software information security assurance	57
B.1 Software safety assurance	57
B.2 Software information security assurance	58
Annex C (informative) Rights in technical data and software usage.....	61

Annex D (informative) Acquisition plan guidelines.....	62
D.1 (AP Section 1) Introduction.....	62
D.2 (AP Section 2) References.....	63
D.3 (AP Section 3) Definitions.....	63
D.4 (AP Section 4) Software acquisition overview	63
D.5 (AP Section 5) Software acquisition process.....	64
D.6 (AP Section 6) Software acquisition reporting requirements.....	64
D.7 (AP Section 7) Software acquisition management requirements.....	64
D.8 (AP Section 8) Software acquisition documentation requirements.....	65
Annex E (informative) Bibliography.....	66

IEEE Recommended Practice for Software Acquisition

IMPORTANT NOTICE: IEEE Standards documents are not intended to ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.

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~~ essential 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~~ describes the options for acquiring 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~~ eight 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~~ describes how quality ~~products and services from~~ assurance can be applied to software ~~suppliers~~ acquisition.

This recommended practice also contains ~~three~~ five 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~~ software acquisition guidelines with respect to Software Safety Assurance and Software Information Security Assurance, Annex C provides guidelines with respect to rights in technical data software usage, Annex D provides acquisition plan guidelines, and Annex E is the bibliography.

1.1 Scope

This ~~is a~~ recommended practice ~~for performing software acquisitions. It~~ describes a set of useful quality ~~practices~~ considerations that can be selected and applied during one or more steps in a software acquisition process. ~~The~~ This recommended practices 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 MOT S~~ The software supply chain may include integration of commercial-off-the-shelf (COTS), custom, or free and fully-developed open source software (FOSS). Each organization or individual 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 organization's acquisition process. Security will be included as a quality attribute considered during the acquisition.

~~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).~~

1.2 Purpose

This recommended practice is designed to help organizations and individuals incorporate quality, including security considerations during the definition, evaluation, selection, and acceptance of supplier software for operational use. It will also help 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 software from software suppliers.

- == [Provide useful practices on including quality, security, safety and data rights 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.](#)
- == [Assist suppliers in understanding how the software will be evaluated, tested, and accepted for delivery to end users.](#)

6. Summary

Success in acquiring high-quality software products and services from software suppliers can be achieved by doing the following things:

- a) Identifying quality characteristics necessary to achieve the acquirer's objectives
- b) [Selecting suppliers most capable of meeting the acquisition objectives](#)
- c) Including quality considerations in the planning, evaluation, and acceptance activities
- d) Developing an organizational strategy for acquiring software

- e) Establishing a software acquisition process using the ~~nine~~ eight steps stated in 5.2 as a starting point
- f) Putting the defined process into practice

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. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 1228™, IEEE Standard for Software Safety Plans.^{1, 2}

¹ IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., 445 Hoes Lane, ~~P.O. Box 1331~~, Piscataway, NJ ~~08855-1331~~ 08854, USA (<http://www-standards.ieee.org/>).

² The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

[ISO/IEC/IEEE Std 12207, Systems and software engineering—Software life cycle processes.](#)

[ISO/IEC/IEEE Std 15288, Systems and software engineering—System life cycle processes.](#)

[ISO/IEC/IEEE Std 15289, Content of life-cycle information products.](#)

[ISO/IEC/IEEE Std 24765, Systems and Software Engineering—Vocabulary.](#)

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

~~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.~~

~~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~~

² ~~ISO publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembe, CH-1211, Geneve 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/>).~~

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

(Deleted)

3. Definitions and terms

For the purposes of this document, the following terms and definitions apply. The *International Standard ISO/IEC/IEEE 24765: Systems and Software Engineering—Vocabulary* should be referenced for terms not defined in this clause.³

³ The *International Standard ISO/IEC/IEEE 24765: Systems and Software Engineering—Vocabulary* is available at www.computer.org/sevocab.

3.1 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.~~

acquisition: ~~The~~ Process of obtaining a system, product, or ~~software~~ [service](#).

commercial-off-the-shelf (COTS): [A product available for purchase and use without the need to conduct development activities \[ISO/IEC/IEEE 15289:2011\].](#) ~~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.~~

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.

off-the-shelf (OTS): Product or system already developed and available [ISO/IEC/IEEE 15289:2011].

software-intensive system: A system for which software is a major technical challenge and is perhaps the major factor that affects system schedule, cost, and risk [IEEE Std 1362-1998 (Reaff 2007)] and [ISO/IEC/IEEE 24765:2010].

NOTE—In the most general case, a software-intensive system is comprised of hardware, software, people, and manual procedures.⁴

~~**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).~~

⁴ Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

~~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.~~

3.2 Use of should, may, and can

In this document, the word *should* is used to indicate a recommendation. The word *may* is used to indicate a permissible action. The word *can* is used for statements of possibility and capability.

4. Software acquisition alternatives

4.1 Introduction

Software can be acquired in three different ways or a combination of the three. It is acknowledged that any combination of these three alternatives will make the overall software project more complex and likely to be different from each other. This document does not attempt to address the combinations of these three basics software acquisition options. The software acquisition alternatives include contracting custom-developed software, obtaining the right to use off-the-shelf (OTS) software and renting software as a service (SaaS). The advantages and disadvantages of each option are summarized in Table 1.

4.2 Custom-developed software

Reaching an agreement, often a formal contract for the custom development is an acquisition approach where the solution to an acquirer's need is provided via hiring a supplier (outsourcing) or using an internal team to custom develop a software solution that is built to a specification or set of requirements. Following software development standards and known methods, the acquirer oversees the development, test activities, and products performed by the supplier. At acceptance, the acquirer takes ownership of the solution and possibly some of the supporting tools/methods in order to support ops/maintenance of the solution. This option includes software built from scratch and/or integrated from other existing components (e.g., reuse, modified FOSS).

4.2.1 Advantages of custom-developed software

The advantages of custom-developed software are as follows:

- The solution should match the need exactly, no more or no less.
- The acquirer can require that flexibility and maintainability be supported by the solution and by the supplier of the solution.
- The acquirer can monitor and influence the course of development/test in case changes are deemed appropriate.
- The acquirer can control and influence the related topics to the executable software solution (e.g., size, performance, information protection, authorization security, capacity) and the software development environment (language, tools, etc.).
- If a problem is reported, the acquirer can determine if the problem legitimately violates the specifications and if it does, the supplier should be responsible for fixing that problem, typically at no additional cost to the acquirer.
- Changes after acceptance are typically still under control of the acquirer via an ops/maintenance agreement with a possibly different ops/maintenance supplier.

4.2.2 Disadvantages of custom-developed software

The disadvantages of custom-developed software are as follows:

- == Typically costs more than the other options.
- == Typically takes more time (sometimes much more) than the other options.

4.3 Off-the-shelf (OTS) software

This is an acquisition approach where the solution to an acquirer's need is provided via purchasing and taking possession of an already built solution from a software supplier. The acquirer agrees to the seller's license agreement to use the "executable" version of the solution but typically does not have access to the source code or other information about the solution or the processes used to produce that product. In addition, the acquirer typically takes possession of the software solution and installs in on assets owned by the acquirer. In addition, the solution is typically acquired in "as is" condition, unless the supplier offers built-in tailoring of the product (and this more common than not). This option includes commercial off-the-shelf (COTS), free and open source software (FOSS) products acquired in "as is" condition, and other non-developmental items (NDI).

4.3.1 Advantages of OTS

The advantages of OTS are as follows:

- == Cost is less than the custom-developed option but similar to the SaaS option.
- == Time to initial operations is less than the custom-developed option and can be greater than the SaaS option.
- == Since the acquirer takes possession of the solution, the control over data and access is greater than the SaaS option and about the same as the custom-developed option.

4.3.2 Disadvantages of OTS

The disadvantages of OTS are as follows:

- == The terms of the generally non-negotiable license agreement constrain how the solution may be used by the acquirer, possibly to the acquirer's detriment. The solution typically includes features that attempt to enforce specific agreement terms, adding cost and complexity to the solution.
- == The solution may not satisfy all of the acquirer's needs (sometimes called gaps) and it may also provide features/capabilities that are not required, not desired or even tolerable to the acquirer. The acquirer may need additional solutions, or modifications to the acquired solution, to fill these gaps. In addition, the acquirer may also need to identify and deactivate or prevent the extraneous code from activating itself.

- == Sometimes, the User Interface to the solution is different from the way an acquirer is currently offering a User Interface to their current users. This can sometimes/often be solved via product tailoring.
- == Typically, there will be a need to obtain support from the supplier for installation, activation, and maintenance.
- == Problem reporting can be an issue, depending on the supplier's capability to accept and react to problem reports. In addition, the supplier may not even agree that because an acquirer alleges a problem exists that a legitimate problem actually does exist.
- == Product changes, both fixes and upgrades, are controlled mostly by the supplier and determined by marketplace needs. Such changes may not be consistent with the acquirer's ongoing needs or timeframe.
- == It is possible for the vendor to go out of business resulting in a possible loss of data.

4.4 Software as a service (SaaS)

Software as a service (SaaS) is an acquisition approach where the solution to an acquirer's need is provided via an agreement with a supplier for access to a software product for a fixed amount of time and to use that software product remotely, via connection (wired, wireless, etc.) to the supplier's assets. Inputs to the remote software product are transmitted to the supplier and outcomes/results/products from using the software product are transmitted back to the acquirer. This acquisition option is part of the Cloud Computing approach (NIST Special Publication 800-145 [B12]) to software (aka renting access to products, services, infrastructure, etc.) and is frequently referred to as software as a service (SaaS).

4.4.1 Advantages of SaaS

The advantages of SaaS are as follows:

- Rapid access to products and services via Service Level Agreement (SLA) and ready to use solutions, more so than custom-developed or OTS.
- No need to provide internal computing assets to support “building or buying” the equivalent products or services, thus reducing costs and time in prep for using solution, as compared to both custom-developed and OTS.
- Typically is cheaper than custom-developed option and can be cheaper than the OTS option.

- No need to take permanent ownership of the software solution, thus reducing impacts associated with such ownership (e.g., sustainment, hardware to host solution), as in the custom-developed and OTS options.

4.4.2 Disadvantages of SaaS

The disadvantages of SaaS are as follows:

- The terms of the generally non-negotiable license agreement constrain how the solution may be used by the acquirer, possibly to the acquirer's detriment. The solution typically includes features that attempt to enforce specific agreement terms, adding cost and complexity to the solution.
- The solution may not satisfy all of the acquirer's needs (sometimes called gaps) and it may also provide features/capabilities that are not required, not desired or even tolerable to the acquirer. The acquirer may need additional solutions, or modifications to the acquired solution, to fill these gaps. In addition, the acquirer may also need to identify and deactivate or prevent the extraneous code from activating itself.
- The location of where the data is stored is not controlled by the acquirer; therefore the data (personal data, sensitive data, etc.) could be subject to the rules and regulations of the locality of the supplier.
- In the event the supplier is no longer a case of e.g., bankruptcy of supplier, potentially the acquirer's data could be lost.

- == The user interface to the solution may be different from the way an acquirer is currently offering a user interface to their current users. This can sometimes/often be solved via product tailoring.
- == Protection of data, being transmitted in both directions due to dependency on supplier's capabilities. This may require internal protections or legal agreements if this protection should ever fail or be compromised.
- == Problem reporting and problem correction of a solution, if it does not work as desired: Supplier may not agree with nature of the problem or may not be able to fix in a timely manner. Similar to OTS and more problematic than custom-developed.
- == Assured access to solution due to dependency on external assets. Again, backup/alternative paths to supplier's assets may be required or legal recourse in case of loss of access. This is typically a bigger risk than for custom-developed or OTS options.
- == Recovery from loss of access to or other failures of the solution may be dependent on the supplier's abilities and sometimes out of the control of the acquirer.
- == Product changes, both fixes and upgrades, are controlled mostly by the supplier and determined by marketplace needs. Such changes may not be consistent with the acquirer's ongoing needs or timeframe.
- == It is possible for the vendor to go out of business resulting in a possible loss of data.

Table 1— Software acquisition options

	Advantages	Disadvantages
Custom-developed software	1. The solution should match the need exactly, no more or no less. 2. The acquirer can require that flexibility and maintainability be supported by the solution and by the supplier of the solution. 3. The acquirer can monitor and influence the course of development and test in case changes are needed. 4. The acquirer can control and influence the related topics to the executable software solution (e.g., size, performance, information protection, authorization security, capacity) and the software development environment (language, tools, etc.). 5. If a problem is reported, the acquirer can determine if the problem legitimately violates the specifications and if it does, the supplier should be responsible for fixing that problem, typically at no additional cost to the acquirer. 6. Changes after acceptance are typically still under control of the acquirer via an operations/maintenance agreement with a possibly different operations/maintenance supplier.	1. Typically costs more than the other options. 2. Typically takes more time than the other options.
Off-the-shelf (OTS) software	1. Cost is less than custom-developed option but similar to the SaaS option. 2. Time to initial operations is less than the custom-developed option but greater than the SaaS option. 3. Since the acquirer takes possession of the solution, the control over data and access is greater than the SaaS option and about the same as the custom-developed option.	1. The terms of the generally non-negotiable license agreement constrain how the solution may be used by the acquirer, possibly to the acquirer's detriment. The solution typically includes features that attempt to enforce specific agreement terms, adding cost and complexity to the solution. 2. The solution may not satisfy all of the acquirer's needs (sometimes called "gaps") and it may also provide features/capabilities that are not required, not desired or even tolerable to the acquirer. The acquirer may need additional solutions, or modifications to the acquired solution, to fill these gaps. In addition, the acquirer may need to identify and deactivate or prevent the extraneous code from activating itself. 3. Sometimes the user interface to the solution is different from the way an acquirer is currently offering a user interface to their current users. This can sometimes/often be solved via product tailoring. 4. Typically, there will be a need to obtain support from the supplier for installation, activation and maintenance. 5. Problem reporting can be an issue, depending on the supplier's capability to accept and react to problem reports. In addition, the supplier may not even agree that because an acquirer alleges a problem exists that a legitimate problem actually does exist. 6. Product changes, both fixes and upgrades, are controlled mostly by the supplier and determined by marketplace needs. Such changes may not be consistent with the acquirer's ongoing needs or timeframe. 7. It is possible for the vendor to go out of business resulting in a possible loss of data.

(New)

	Advantages	Disadvantages
Software as a service (SaaS)	1. Rapid access to products and services via Service Level Agreement (SLA) and ready-to-use solutions, more so than custom-developed or OTS. 2. No need to provide internal computing assets to support 'building or buying' the equivalent products or services, thus reducing costs and time in preparation for using solution, as compared to both custom-developed and OTS. 3. Typically is cheaper than custom-developed option and can be cheaper than the OTS option. 4. No need to take permanent ownership of the software solution, thus reducing impacts associated with such ownership (e.g., sustainment, hardware to host solution), as in the custom-developed and OTS options.	1. The terms of the generally non-negotiable license agreement constrain how the solution may be used by the acquirer, possibly to the acquirer's detriment. The solution typically includes features that attempt to enforce specific agreement terms, adding cost and complexity to the solution. 2. The solution may not satisfy all of the acquirer's needs (sometimes called gaps) and it may also provide features/capabilities that are not required, not desired or even tolerable to the acquirer. The acquirer may also need to identify and deactivate or prevent the extraneous code from activating itself. 3. Sometimes the user interface to the solution is different from the way an acquirer is currently offering a user interface to their current users. This can sometime/often be solved via product tailoring. 4. Typically, there will be a need to obtain support from the supplier for installation, activation, and maintenance. 5. Problem reporting and problem correction of a solution, if it does not work as desired. 6. Recovery from loss of access to or other failures of the solution may be dependent on the supplier's abilities and sometimes out of the control of the acquirer. 7. Product changes, both fixes and upgrades, are controlled mostly by the supplier and determined by marketplace needs. Such changes may not be consistent with the acquirer's ongoing needs or timeframe. 8. It is possible for the vendor to go out of business resulting in a possible loss of data.

(New)

5. ~~Introducing the~~ Software acquisition process

5.1 Purpose

The purpose of the acquisition process is to obtain a product or service in accordance with the acquirer's requirements.

5.1.1 Outcomes

As a result of the successful implementation of the acquisition process:

- a) A request for supply is prepared.
- b) One or more suppliers are selected.
- c) An agreement is established between the acquirer and supplier.
- d) A product or service complying with the agreement is accepted.
- e) Acquirer obligations defined in the agreement are satisfied.

5.1.2 Activities and tasks

Software can be acquired as a stand-alone product, as a service, or embedded in a system consisting of a single processor or a large, complex system with many hardware items. This clause describes the activities in the software acquisition process and is applicable to the acquisition of developed software, non-developmental software (COTS, reuse, or FOSS), and software services.

5.1.3 Principal acquisition roles

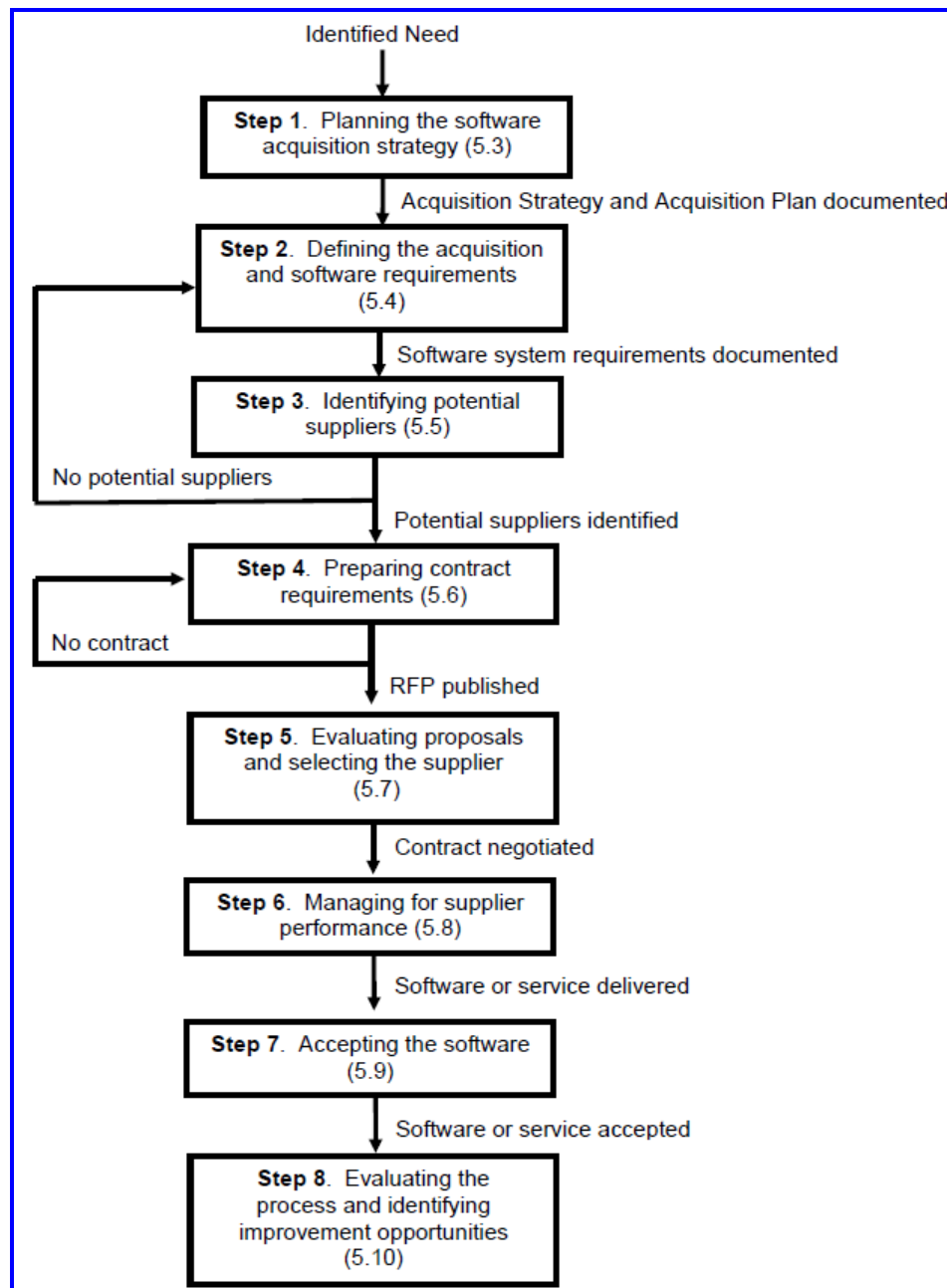
There are three principal roles involved: acquirer, supplier, and end user. These roles may be performed by a single individual in a small business or by multiple people representing organizations in a large corporate or government environment. The acquirer develops the method to satisfy the identified need and the supplier provides the required product or service to the end user.

5.2 Eight steps in acquiring quality software

The software acquisition process can be described by eight steps. The activities in each step support the acquisition of high-quality software products, services or systems that meet the acquirer's requirements and expectations, including on-time and on-budget delivery.

The software acquisition process should be tailored or streamlined to the maximum extent possible taking risk into consideration. The process should be consistent with technical maturity, validated requirements, and expected level of effort and funding. The tailoring process is presented in Annex A of ISO/IEC/IEEE 12207.

Figure 1 illustrates the eight-step process described in this clause and Table 2 summarizes the process steps, along with their inputs and outputs.



(New)

Figure 1 – Eight steps in acquiring software

Table 2— Eight steps in acquiring software

Steps in software acquisition process	Activities and Tasks	Outcomes
Step 1. Planning the software acquisition strategy (5.3)	1. Initiate a planning process. 2. Develop an acquisition strategy. 3. Develop an acquisition plan. 4. Include contracting practices. 5. Obtain acquisition services from other organizations. 6. Tailor the process.	1. High level specification of functional and non-functional requirements to confirm need for a system and guide acquisition planning is produced. 2. Acquisition plan is specified and deliberated on. 3. Acquisition plan is approved.
Step 2. Defining the acquisition and software requirements (5.4)	1. Define the software being acquired. 2. Establish supplier evaluation criteria. 3. Establish acquirer and supplier obligations. 4. Develop plans to evaluate and accept software and services. 5. Develop contingency plans.	1. The software to be acquired is defined in detail. 2. The Statement of Work (SOW) is defined and approved. 3. Software requirements are specified and approved. 4. A detailed RFP which includes acquirer and supplier obligations is produced and approved. 5. Evaluation and acceptance criteria for the software to be acquired are produced. 6. Contingency plan is drawn up and approved.
Step 3. Identifying potential suppliers (5.5)	1. Gather software supplier's information. 2. Review software supplier's information. 3. Survey uses of the supplier's software.	1. Potential supplier information is gathered and a detailed RFI report produced for each supplier visited. 2. Review report based on potential supplier's information is produced. 3. A limited number of supplier candidates are shortlisted based on the acquisition organization's policy.
Step 4. Preparing contract requirements (5.6)	1. Determine the required quality of the work. 2. Determine how payment is to be made. 3. Determine nonperformance remedies. 4. Prepare contract provisions. 5. Review contract provisions with legal counsel.	1. The desired level of quality of work is determined. 2. Payment terms are determined. 3. Remedies for non-performance are determined. 4. Contract for engaging a supplier is produced and approved. 5. Legal review opinion including caveats and limitations documented.
Step 5. Evaluating proposals and selecting the supplier (5.7)	1. Evaluate supplier proposals. 2. Visit supplier facilities. 3. Select a qualified supplier. 4. Negotiate the contract.	1. An evaluation report on proposals submitted is produced. 2. A qualified and preferred supplier is selected. 3. A contract between acquirer and supplier is negotiated and signed.
Step 6. Managing for supplier performance (5.8)	1. Manage the contract during execution. 2. Monitor the supplier's progress.	1. Performance measures or assessment results are available. 2. Adequacy of roles and accountabilities is assessed. 3. Agreement objectives are achieved. 4. Accountability in carrying out project activities and tasks. 5. Issues (including problems) in project execution are managed.
Step 7. Accepting the software (5.9)	1. Review and update the acceptance criteria. 2. Conduct a review and test of the software. 3. Manage the testing process. 4. Document the test results.	1. An acceptance testing report on delivered software solution is produced. 2. Decision to accept, reject or make changes to the delivered software solution is made.
Step 8. Evaluating the process and identifying improvement opportunities (5.10)	1. Evaluate contracting practices. 2. Evaluate software product quality. 3. Evaluate supplier performance.	1. Contracting practices evaluated and reviewed. 2. Acquired software product is evaluated. 3. Supplier performance is evaluated. 4. A detailed evaluation report is produced.

(New)

5.3 Step 1: Planning the software acquisition strategy

5.3.1 Purpose

- a) The purpose of planning the software acquisition is to define the acquisition strategy. Planning for software acquisition begins when an organization has established a need for a software system and has made the decision to acquire the software rather than to develop it in-house. This includes the activities performed to define an acquisition strategy for the software to be acquired. The acquisition strategy includes a description of what is to be acquired, the length of time estimated for implementation, and an estimate of the cost of the implementation. The acquisition strategy also includes a concept for how the software will be operated and maintained; whether by the acquirer, the supplier, or by a third party.
- b) Planning includes the definition of the acquisition plan to be used by the acquirer. The acquisition plan describes the activities that the acquisition organization should perform to acquire the software.

5.3.2 Outcomes

The outcomes of Step 1 are as follows:

- High level specification of functional and non-functional requirements to confirm need for a system and guide acquisition planning is produced.

— [Acquisition plan is specified and deliberated on.](#)

— [Acquisition plan is approved.](#)

5.3.3 Activities and tasks

5.3.3.1 Initiate a planning process

[Before initiating planning, the acquirer should ascertain if any established acquisition processes policies exist \(preferred supplier lists, etc.\) for the planning organization.](#)

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 should possess to achieve the organization's objectives

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

(Deleted)

^a The step numbers referred to in this table correlate to the nine steps outline 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.~~

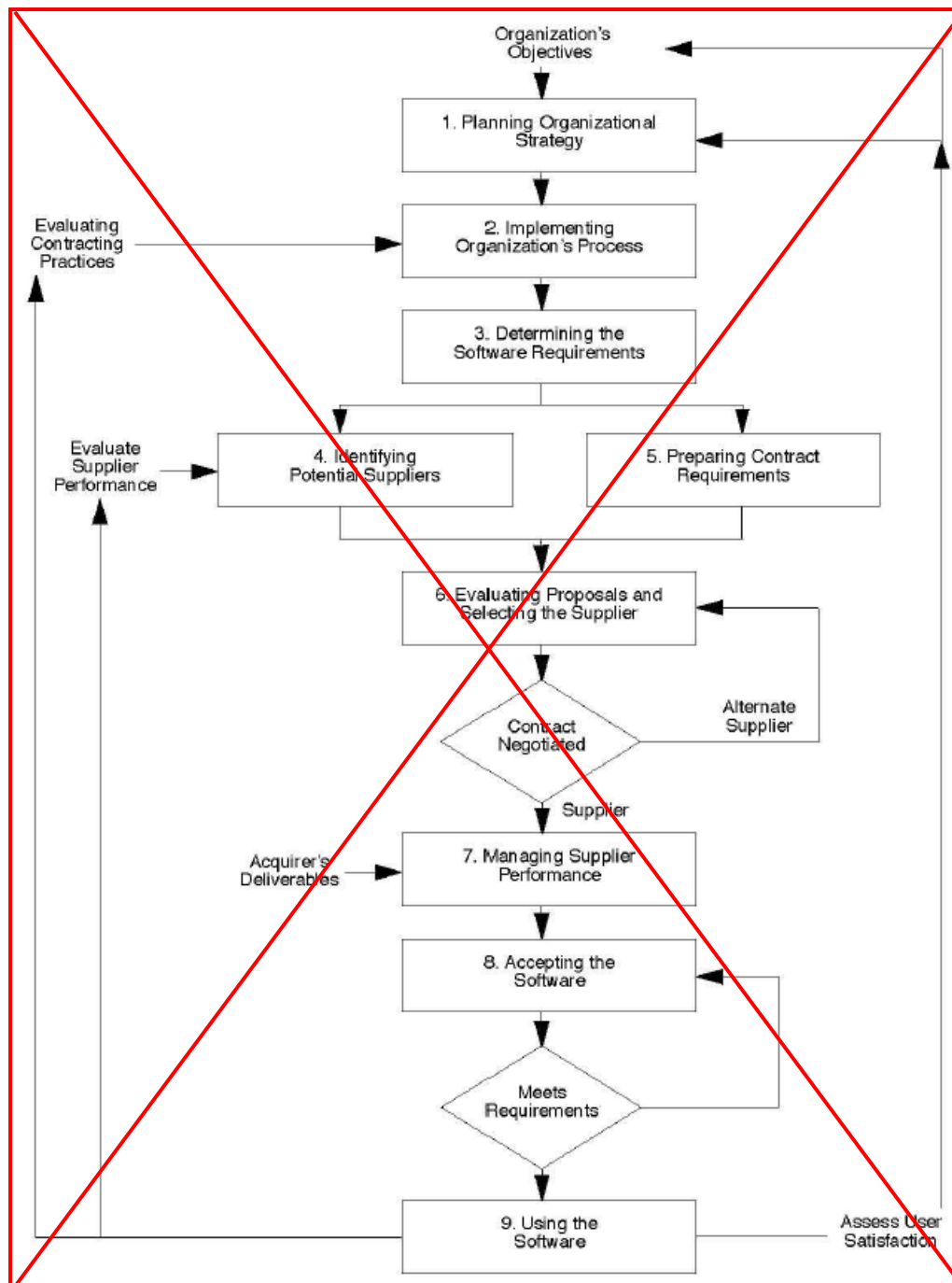
~~Step 6: Evaluating proposals and selecting the supplier. Evaluate supplier proposals, select a qualified supplier, and negotiate the contract. Negotiate with an alternate supplier, if necessary.~~

~~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.~~



(Deleted)

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.