IEEE Std 1462-1998™
(Adoption of International Standard ISO/IEC 14102: 1995)

Information technology—Guideline for the evaluation and selection of CASE tools

Sponsor

Software Engineering Standards Committee
of the
IEEE Computer Society

Adopted 19 March 1998 by

IEEE-SA Standards Board

Abstract: IEEE Std 1462-1998 is an adoption of International Standard ISO/IEC 14102: 1995. The International Standard deals with the evaluation and selection of CASE tools, covering a partial or full portion of the software engineering life cycle. The adoption of the International Standard by IEEE includes an implementation note, which explains terminology differences, identifies related IEEE standards, and provides interpretation of the International Standard.

Keywords: CASE tools, computer-aided software engineering, computer applications, computer software, software engineering

IEEE®
Networking the World™

ISBN 0-7381-0190-7

SH94620

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.

Implementation note

(This implementation note is informative for IEEE Std 1462-1998, Adoption of International Standard ISO/IEC 14102: 1995, Information technology—Guideline for the evaluation and selection of CASE tools. It is not part of ISO/IEC 14102: 1995.)

IEEE Std 1462-1998 is an adoption of ISO/IEC 14102: 1995[1]. The following implementation notes relate to IEEE interpretation of ISO/IEC 14102.

Terminology

The terms "metric" and "measurement" are both used in the document, and can be read as having the same meaning.

Some differences exist between the terminology used in IEEE Std 1462-1998 and that in IEEE Std 610.12-1990. The reader should use the local definitions in IEEE Std 1462-1998 where there is a conflict with those of IEEE Std 610.12-1990. In some cases, the definition in IEEE Std 610.12-1990 may provide further clarification. It is important to keep in mind, however, that full compatibility of definitions has yet to be achieved in the evolving field of software engineering.

Related IEEE standards

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

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

The role of the standard

IEEE Std 1462-1998 deals with the technical evaluation of a CASE tool. This represents only a portion of the overall process that covers the introduction of a new CASE tool into the organization. For example, the business case including the business objectives that the CASE tool is intended to address are part of activities that would precede the use of IEEE Std 1462-1998.

Specifically, IEEE Std 1348-1995 covers the broader view, and places the evaluation as one of four steps. The reader is advised to review IEEE Std 1348-1995 together with IEEE Std 1462-1998 in order to establish a proper scope and business case for the evaluation. IEEE recommends that the evaluation of a CASE tool be treated as a project with its associated budget and schedule controls.

Section 9 and CASE tool characteristics

Section 9 of IEEE Std 1462-1998 describes a set of product characteristics against which the evaluation should be done. These are organized under the following four areas, which together form the basis for establishing the requirements and performing the final evaluation:

1) How the CASE tool meets the needs of the life cycle processes used;

[1] ISO and ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse. 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.

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

2) How the CASE relates to the overall tool environment;
3) How the CASE tool itself performs; and
4) Other criteria.

The requirements are organized as characteristics and subcharacteristics, adopting the software product quality model as defined in ISO/IEC 9126: 1991, *Information technology—Software product evaluation—Quality characteristics and guidelines for their use*. The high-level characteristic (also referred to as quality factor) is subdivided into subcharacteristics, that in turn may be further subdivided. At the lowest level, they are referred to as atomic subcharacteristics. The evaluation of each chosen characteristic and subcharacteristic is done through the establishment of a value for each atomic subcharacteristic.

The user of IEEE Std 1462-1998 should select from the set of characteristics noted. However, the following should be noted:

— Characteristics from each of the four areas should be selected to cover each of the major tool features.
— The characteristics, once selected, should be prioritized. This is particularly true for Areas 1 and 3. These priorities should reflect the main objectives of the tool.

Compliance

IEEE Std 1462-1998 provides guidance in the steps recommended to undertake the evaluation of a CASE tool or tools, and in the features (characteristics) of the CASE tool. The four processes described are not mandatory; there are few “shall” statements, and the tailoring of these processes should be done carefully, since tailoring down may introduce added risk for a CASE evaluation project. Assumptions should be verified with experienced staff whenever possible.

IEEE recommends that any organization undertaking one or more tool evaluations should provide the general level of compliance that it requires for any tool evaluation project. The process activities for a given project can then be further tailored, since the number of activities may vary depending on the criticality of the CASE tool being evaluated.

In situations where the evaluation is being subcontracted, it is recommended that the organization (purchaser) formally specify the minimum processes, activities, and tasks that are to be followed in IEEE Std 1462-1998.

Participants

The following persons were on the balloting committee:

Syed Ali
Pamela K. Arya
Robert E. Barry
Leo Beltracchi
Mordechai Ben-Menachem
H. Ronald Berlack
Alan L.Bridges
Kathleen L. Briggs
Edward R. Byrne
Michael Caldwell
Thomas G. Callaghan
Keith Chan
John P. Chihorek
Antonio M. Cicu
Theo Clarke
Gregory T. Daich
Bostjan K. Derganc
Sanjay Dewal
Leo Egan
William Eventoff
Jonathan H. Fairclough
John W. Fendrich
Jay Forster
Simon Gabrihelidis
Juan Garbajosa-Sopena
Hiranmay Ghosh
Marilyn Ginsberg-Finner
John Garth Glynn
Julio Gonzalez-Sanz
Lewis Gray

L. M. Gunther
John Harauz
Herbert Hecht
Manfred Hein
Mark Heinrich
Mark Henley
Jerry Huller
Fabrizio Imelio
Frank V. Jorgensen
Vladan V. Jovanovic
William S. Junk
Ron S. Kenett
Judith S. Kerner
Robert J. Kierzyk
Dwayne L. Knirk
Shaye Koenig
Thomas M. Kurihara
John B. Lane
J. Dennis Lawrence
Randal Leavitt
Michael Lines
William M. Lively
Dieter Look
Philip P. Mak
Tomoo Matsubara
Sue McGrath
Bret Michael
Alan Miller
Charles S. Mooney
James W. Moore
R. Muralidharan

Pavol Navrat
Myrna L. Olson
Mike Ottewill
Indradeb P. Pal
Lalit M. Patnaik
Donald J. Pfeiffer
Alex Polack
Peter T. Poon
Kenneth R. Ptack
John Reddan
Ann E.Reedy
Andrew P. Sage
Helmut Sandmayr
Stephen R. Schach
Norman Schneidewind
Robert W. Shillato
Carl A. Singer
Richard S. Sky
Alfred R. Sorkowitz
Julia Stesney
Fred J. Strauss
Toru Takeshita
Leonard L. Tripp
Richard D. Tucker
Thomas E. Vollman
Dolores Wallace
David M. Waxman
Camille S. White-Partain
Scott A. Whitmire
Peter F. Zoll

When the IEEE-SA Standards Board adopted this standard on 19 March 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

INTERNATIONAL STANDARD

ISO/IEC 14102

First edition
1995-11-15

Information technology — Guideline for the evaluation and selection of CASE tools

Technologies de l'information — Lignes directrices pour l'évaluation et la sélection d'outils CASE

Reference number
ISO/IEC 14102:1995(E)

Contents

ISO/IEC Copyright Office • Case Postale 56 • CH-1211 Geneve 20 • Switzerland
Printed in Switzerland

Foreword

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

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 14102 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* Subcommittee SC 7, *Software engineering.*

Annexes A to C of this International Standard are for information only.

Introduction

Within software engineering, Computer-Aided Software Engineering (CASE) tools represent a major part of the supporting technologies used to develop and maintain software systems. Their selection must be carried out with careful consideration of both the technical and management requirements.

This International Standard defines both a sequence of processes and a structured set of CASE tool characteristics for use in the technical evaluation and the ultimate selection of a CASE tool. It follows the software product evaluation model described in ISO/IEC 9126:1991, *Information technology - Software product evaluation - Quality characteristics and guidelines for their use*. This International Standard adopts the general model of software product quality characteristics and subcharacteristics described in ISO/IEC 9126, and extends these when the software product is a CASE tool; it provides product characteristics unique to CASE tools. This larger set of characteristics is then organized into five groups. This grouping provides a more manageable approach to the overall evaluation and selection process.

While the technical evaluation may answer how well a CASE tool meets its user's stated requirements, it may also answer the question of how well the tool meets its claimed functionality.

The objective of the technical evaluation process is to provide quantitative results on which the final selection can be based. Measurement assigns numbers (or other ratings) to attributes of entities; a major activity of evaluation is to obtain these measurements for use in selection. The final selection results should aim to achieve objectivity, repeatability and impartiality. These objectives and the confidence in the outcomes will in part depend on the resources allocated to the overall evaluation and selection process. The user of this International Standard is asked to deal with these issues at an early stage.

To be widely acceptable, these CASE tool evaluation and selection processes must be of value to the users of CASE tools, to the suppliers of CASE to community at large. The information outlined in this International Standard should lead to more cost effective selections of CASE tools and to a greater uniformity in how CASE tool functions and features are described.

Information technology - Guideline for the evaluation and selection of CASE tools

1 Scope

This International Standard deals with the evaluation and selection of CASE tools, covering a partial or full portion of the software engineering life cycle. It establishes processes and activities to be applied for the evaluation of CASE tools and selecting the most appropriate CASE tools from several candidates. These processes are generic, and organizations must tailor them to meet organizational needs. The CASE tool evaluation and selection processes should be viewed in the larger context of the organization's technology adoption process.

This International Standard provides:

a. Guidance on identifying organizational requirements for CASE tools.

b. Guidance on mapping those requirements to CASE tool characteristics to be evaluated.

c. A process for selecting the most appropriate CASE tool from several tools, based on measurements of the defined characteristics.

Primary users of this International Standard are organizations that intend to adopt CASE tools to support their software life cycle processes. CASE tool suppliers may also use this International Standard to describe characteristics of their CASE tools.

This International Standard is not intended to apply to:

a. Software engineering frameworks whose purpose is to provide mechanisms for data, control and presentation integration.

b. General purpose tools (e.g., word processors, spreadsheets) which may be used in software engineering activities, nor CASE tools of very narrow scope or specific purpose (e.g., a compiler).

c. Planning for the implementation of CASE tools within an organization (even though it is recognized that this is an important subject).

NOTE - A user of this International Standard may make the best possible selection of a CASE tool and have no guarantee of a successful implementation. ISO/IEC JTC1 SC7 WG4 is working on a draft technical report, Adoption of CASE Tools, which addresses this subject.

This International Standard contains a set of processes, activities, and tasks designed to be tailored. The tailoring process is the selection of applicable processes, activities and tasks.

Compliance with this International Standard is defined as the performance of the processes, activities, and tasks selected from this International Standard for the evaluation and selection project. Any organization imposing this International Standard as a condition of trade is responsible for specifying the minimum set of required processes, activities, and tasks which constitute compliance for a given application of this International Standard. Defining and documenting that specification forms part of the initiation process (clause 5).

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreement based upon this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 5807:1985, *Information processing - Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts.*

ISO/IEC 12119:1994, *Information technology - Software packages - Quality requirements and testing.*

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

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

3 Definitions and acronyms

3.1 Definitions

For the purposes of this International Standard, the following definitions apply.

3.1.1 assessment: An action of applying specific documented criteria to a specific software module, package or product for the purpose of determining acceptance or release of the software module, package or product. (ISO/IEC 9126:1991)

3.1.2 atomic subcharacteristic: The highest level evaluation categories are called characteristics. Characteristics are usually subdivided into subcharacteristics. Many subcharacteristics may be further subdivided into lower level subcharacteristics. At the lowest-level, when no further subdivision is appropriate, the subcharacteristics are referred to as atomic subcharacteristics.

3.1.3 CASE tool: A software product that can assist software engineers by providing automated support for software life-cycle activities as defined in ISO/IEC 12207:1995.

NOTES

1 - A CASE tool may provide support in only selected functional areas or in a wide variety of functional areas.

2 - CASE tools may be used in several modes:

- As stand alone tools; in this case, only compatibility with environment elements should be addressed.
- In small groups which communicate directly with one another; it may be supposed that integration is predefined, perhaps proprietorily.
- In the presence of a larger framework of the SEE; in this case the ability of the tool to use the relevant services of the framework should be addressed.

3.1.4 characteristic: An aspect of a product by which it can be described and evaluated. A characteristic may be refined into multiple levels of subcharacteristics that bear on its ability to satisfy stated or implied needs.

3.1.5 measurement: The action of applying a software quality metric to a specific software product. (ISO/IEC 9126:1991)

NOTES

1 - Measurement can apply to metrics other than software quality metrics.

2 - An object may be measured directly, or may be measured indirectly by the application of metrics to information about or representations of the object.

3.1.6 metric: A quantitative scale and method which can be used to determine the value a subcharacteristic takes for a specific software product.

3.1.7 rating: The action of mapping the measured value to the appropriate rating level. Used to determine the rating level associated with the software for a specific quality characteristic. (ISO/IEC 9126:1991)

NOTE - Rating and rating levels can be applied to characteristics other than quality characteristics.

3.1.8 rating level: A range of values on a scale to allow software to be classified (rated) in accordance with the stated or implied needs. Appropriate rating levels may be associated with the different views of quality, i.e., users, managers or developers. These levels are called rating levels. (ISO/IEC 9126:1991)

3.1.9 Software Engineering Environment: The software engineering environment (SEE) is that portion of the system which provides automated support for the engineering of software systems and the management of the software process. It includes platform, system software, utilities, and CASE tools installed.

NOTE - The SEE architecture has two aspects:

- the CASE tools which provide facilities for supporting life-cycle processes, and
- a general framework which provides a set of capabilities that offer common services used by the tools.

3.2 Acronyms

BMT	Bench Mark Test
CASE	Computer Aided Software Engineering
GUI	Graphical User Interface
SEE	Software Engineering Environment
SQL	Structured Query Language

4 Overview of evaluation and selection of CASE tools

This section provides an overview of the evaluation and selection of CASE tools discussed in this International Standard as shown in Figure 1. Evaluation and selection of CASE tools includes four major processes:

Initiation Process
Structuring Process
Evaluation Process
Selection Process

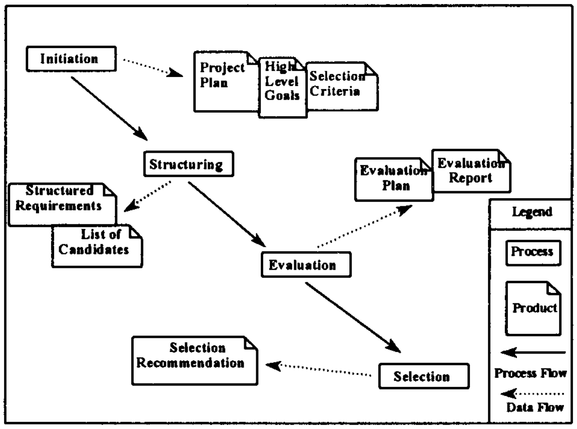

Figure 1 - Overview of evaluation and selection of CASE tools

A key process is the structuring of a set of requirements against which candidate CASE tools are to be evaluated, and upon which selection decisions will be based. The CASE tool characteristics defined in clause 9 form the basis for requirements structuring, and play a central role in the overall process.

4.1 Initiation process

The purpose of the initiation process is to define the general objectives and requirements of the intended evaluation and selection of CASE tools, to establish the high level direction, and to define the management aspects of the effort (e.g., schedule, resources, cost).

The initiation process, discussed in detail in clause 5, is composed of three activities:

- **goal setting**: provides the rationale and general policy for evaluation and selection.

- **establishing selection criteria**: provides criteria to be used in the subsequent selection process.

- **project planning**: results in a plan which includes generic planning information and also information which defines the structure of the evaluation and selection effort.

4.2 Structuring process

The purpose of the structuring process is to elaborate a set of structured requirements, based upon the CASE tool characteristics of clause 9 against which CASE tools should be evaluated, and to obtain the necessary information on CASE tools to permit evaluation. It is assumed that a set of general organizational information and guidelines is available to be used as inputs.

The structuring process, discussed in detail in clause 6, is composed of three activities:

- **requirements analysis**: transforms organizational needs into measurable structures.

- **CASE tool information gathering**: captures a snapshot of the current state-of-the-art in CASE tools.

- **identifying final candidate CASE tools**: candidate CASE tools are identified for evaluation using the results of the last two activities.

NOTE - During the evaluation, requirements may require revision. If this occurs, some repetition of activities of this, and subsequent processes may be necessary.

4.3 Evaluation process

The purpose of the evaluation process is to produce technical evaluation reports that will be the major input for the selection process. Each evaluation process results in a profile of the quality and other characteristics of the tool which was evaluated. Comparisons between tools are not made as part of this process.

The evaluation process, discussed in detail in clause 7, is composed of three activities:

- **preparation for evaluation**: finalization of the various details of the evaluation (e.g., scenario, subcharacteristics, metrics, tool characteristics) in an evaluation plan.

- **evaluating CASE tools**: measurement, rating and assessment.

- **evaluation reporting**: an evaluation report is prepared which provides the results of the evaluation for each CASE tool considered.

4.4 Selection process

The purpose of the selection process is to identify the most suitable CASE tool(s) among the candidate tools, and to ensure that the recommended tool(s) meets the original goals. The selection process compares the results of the evaluations of the candidate tools to determine which is the most appropriate for selection.

The selection process, discussed in clause 8, is composed of four activities:

- **preparing for selection**: the selection criteria are finalized and the selection algorithm is defined.

- **assessing the evaluation results**: the selection algorithm is applied to the evaluation results.

- **recommending a selection decision**: the best of the candidates is determined.

- **validating the selection decision**: the recommended selection is validated against the original goals.

4.5 General process considerations

There are several considerations that apply to the processes described in this International Standard on a global basis. The intent is for the user of this International Standard to tailor its application in such a way as to maximize the probability of a successful evaluation and selection process, and minimize its cost and risk.

4.5.1 Sequencing of processes

This International Standard does not impose the sequence of process activities described above and in the following sections. It is up to the organization to select the relevant processes and activities needed to meet its evaluation and selection goals. The organization will decide which to employ, in what sequence, and with what degree of parallelism. The sequencing of the processes' activities is then documented in the project plan developed during the initiation process.

4.5.2 Reducing cost and risk

In general, organizations which apply this International Standard will want to minimize the cost of the entire evaluation and selection process to the extent possible, while maintaining the level of effort necessary to select the most appropriate CASE tool(s) for their use. These objectives may be addressed by minimizing the number of tools evaluated, minimizing the cost of evaluating specific tools,.and ensuring that the formality of the process is appropriate to the organization.

The activities of CASE tool information gathering and identifying final candidates for selection (see clause 6) effectively allow the user of this International Standard to screen the available tools against the organization's needs, and eliminate from consideration tools which do not, or are not likely to, substantially address the organization's needs.

NOTE 1 - It may be that the organization is unable to find any tool which appear likely to sufficiently meet its needs. In such a case, the stated needs themselves should be re-examined, and if they are found to accurately reflect the organization's actual requirements for technology improvement, the overall evaluation and selection process may be abandoned. Similarly, if the final candidate tools appear to be marginal in addressing the organization's needs, the level of detail and formality of the subsequent activities should be made to reflect the risk factor, and the organization should be prepared to not select a tool if the evaluation process so indicates, as the typical cost of bringing a new tool into operational use is substantial.

Evaluations of candidate tools may have already been performed and be available to the organization. Such information may be used to reduce the cost of candidate tool evaluation.

NOTE 2 - Previous evaluations which have been performed on a different version of the candidate tool may still yield useful information. Similarly, evaluations which addressed a different set of organizational needs may still provide useful information.

This International Standard calls for the development of several plans and reports, and implicitly, for their review by various personnel within the organization. In addition, activities are required to perform the four processes outlined. The format and level of detail of the data products is left to the discretion of the organization, as is the level of effort necessary to perform the activities.

NOTE 3 - Some organizations may need to limit the scope, detail and formality of the processes to apply this International Standard within existing resource constraints.

5 Initiation process

The evaluation and selection processes require the agreement of management. In line with this agreement, a set of goals for the introduction (or enhancement) of CASE technology will be established. A set of CASE tool selection guidelines will be identified and a project plan developed. The process is shown in Figure 2.

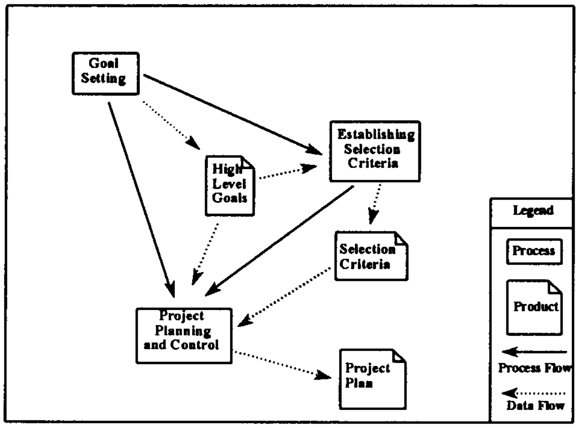

Figure 2 - Overview of initiation process

5.1 Goal setting

The development of a set of realistic goals is a necessary first activity. In developing goals, both a rationale for acquisition (why acquire a CASE tool) and a general policy for acquisition (what type of tool to acquire and how to do it) should be developed.

NOTE - Goal setting activities, including possibly the identification of selection criteria, may have already been performed as a part of other efforts prior to formally entering the initiation process of evaluation and selection of CASE tools.

The following tasks should be performed:

Develop rationale for acquisition:

Review the organization's current software development process, determining its maturity and areas of concern.

Review the current state of CASE technology and observe trends for consideration as future reference technology.

Compare the organization's current practices to possible future practices if

CASE tools are adopted and identify areas of potential benefit.

Identify probable impacts of CASE tools on the organization; e.g., areas where training and education, procedure guides, and technical support are needed to effectively deploy CASE technology.

Define goals and expectations:

Set overall goals (e.g., productivity improvement, quality improvement, enhanced process manageability).

Define evaluation and selection constraints (e.g.,cost, schedule, resources).

Quantify and classify expectations (based upon goals).

Set general policy for acquisition:

Identify constraints on tool acquisition (e.g., implementation cost, schedule, other resources).

Develop alternate approaches to introducing/augmenting CASE technology (e.g., buy a tool, modify an existing tool, develop a new tool).

Assess the feasibility of the various alternatives in light of organizational readiness, technical considerations, performance specifications, and resources.

The goals and expectations established here will be used to guide subsequent activities in the overall process and, finally, to validate the selection decision.

5.2 Establishing selection criteria

Based upon the goals and expectations developed above, selection criteria should be established:

Decompose the high level goals into a set of selection criteria to make the (go/no go) selection decision.

NOTE 1 - The selection criteria should be objective and quantitative. Each selection criterion should include some defined threshold specified on which the major go/no go decision will be made during selection.

Define the relative importance of the selection criteria.

NOTE 2 - The relative importance of the selection criteria will be used to determine the weights assigned to tool characteristics and subcharacteristics for evaluation.

Define the level of detail and the nature of the evaluation activities to be performed.

NOTE 3 - The nature of the evaluation activities covers the methods used in collecting the data. Reference, for example, how the data are measured, collected with predefined criteria, or based upon subjective observation.

Define the evaluation/selection scenario to be performed (see Annex A).

5.3 Project planning and control

Based upon the goals and selection criteria which have been established for the overall evaluation and selection process, a project plan should be created and a control mechanism implemented. The plan and control mechanism should be developed in accordance with the organization's normal planning and control process, and it should contain the following:

A project team organization with assigned responsibilities.

NOTE - The skill of the evaluators will have an impact on the results of the evaluation and its applicability to the organization. The evaluation personnel should be selected with this in mind, and the skill level of evaluators should be a factor in assessing evaluation results. The evaluation team should be representative of the intended tool user group.

A set of operational goals obtained by decomposing the overall goals previously established.

A set of selection guidelines: weighted selection criteria, definition of level of detail and nature, and an evaluation and selection scenario (see Annex A).

A schedule of activities and their tasks, along with an estimate of resource requirements and a cost estimate.

A means of monitoring and controlling the execution of the plan.

If developed, the project plan and control mechanism should be updated as the project evolves.

6 Structuring process

The structuring of the evaluation and selection activities can begin when a set of high level goals, selection guidelines, and a project plan are in place. The structuring process begins with a requirements definition activity which is followed by two parallel activities: the gathering of information on existing CASE tools, and the preparation of a list of candidate CASE tools to be evaluated.

The organization of CASE tool requirements will follow the four groups of CASE tool characteristics as outlined in clause 9. The major activities are shown in Figure 3.

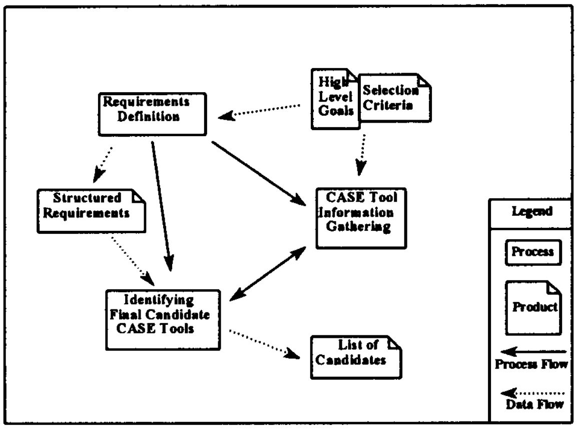

Figure 3 - Overview of structuring process

6.1 Requirements definition

During requirements definition, the requirements for the CASE tool are collected and organized into the CASE tool characteristics as noted in clause 9. 9.1 and 9.2 identify the major CASE specific characteristics; 9.3 identifies general software quality characteristics, and 9.4 identifies a set of characteristics not related to quality. A comprehensive set of requirements is necessary to select the most appropriate CASE tool, and the structuring process provides for greater ease and repeatability in the evaluation process. Three activities are required.

6.1.1 Organizational information gathering

To be able to define a set of detailed requirements to be satisfied by the CASE tool, information about the organization should be gathered, including:

Willingness of the organization to fully fund and implement CASE tool use.

Current software engineering environment within the organization, including data describing current hardware, operating software, and tool use.

Types of software development projects undertaken by the organization include size, domain of application.

Characteristics and constraints of the target systems for which software is developed.

Specific expected impacts and improvements of CASE technology on the organization.

Requirements from potential tool users and end users.

Current organizational procurement policies.

This information is necessary to ensure the tool or tools are appropriate for use within the organization, they address organizational needs, and needs perceived by their future users.

NOTE - This information can be gathered in a number of ways, including surveys and focus groups.

6.1.2 Requirements identification

The tool user's requirements should deal with the question of what the CASE tool should do as well as its impact on the existing environment. The following tasks should be performed in building the list of requirements:

Analyze the requirements and adjust the level of detail to which requirements are defined and measured.

Evaluate the current need for CASE tools while taking into consideration those projects where the CASE tool may initially be used.

Identify desired methodology (e.g., process-oriented, data-oriented, object-oriented).

Identify portions of the life-cycle to be supported (e.g., planning, analysis, design).

Identify required functions of the CASE tool.

Identify required quality characteristics of the CASE tool.

Check the consistency of the requirements with the previously established goals.

NOTE - These requirements represent the total set of organizational requirements. It is possible that no single CASE tool may satisfy all of the requirements, but that individual CASE tools may satisfy a sufficient number to justify their use by the organization, which may continue to search for tools to support remaining requirements.

6.1.3 Requirements structuring

The applicability of the user needs identified in clause 9, and any others which the organization may wish to add, should be defined. The purpose of this structuring is to organize the requirements in such a way that the evaluation can proceed more effectively. The tasks include:

Categorize the user requirements in terms of the organization of clause 9, and decompose them into detailed specifications.

Select characteristics and specific subcharacteristics from clause 9 which can be evaluated to determine the extent to which the CASE tool meets the detailed specifications.

NOTE 1 - The extent to which a CASE tool supports or implements a specific methodology may be a critical factor, and should be seriously considered when selecting characteristics and subcharacteristics and weighting those subcharacteristics.

Identify weights for the characteristics and subcharacteristics.

NOTES

2 - The weights are applied to the ratings determined during the evaluation as part of the selection process, and reflect the relative importance of the related selection criterion as determined during the initiation process.

3 - The assignment of weights is a subjective task which has a fundamental impact on the outcome of the entire evaluation and selection process. The assignment of weights should reflect both the organization's actual requirements and the ability of the organization to evaluate the characteristic. See Annex B for further discussion.

4 - ISO/IEC 12119:1994 addresses quality requirements applicable to CASE tools when considered as software packages, and should be consulted as part of the requirements structuring task. It provides additional guidance on a subset of the quality requirements of ISO/IEC 9126:1991.

6.2 CASE tool information gathering

A general search of potential CASE tools to be evaluated is undertaken based upon the requirements and selection criteria established. The activities of gathering information and identifying the candidate CASE tools may require several iterations to quickly and efficiently identify the most promising tools for further evaluation. For the CASE tools which appear most promising for further evaluation, additional and more detailed data that deal with their potential acquisition are obtained. This additional information may help to quickly eliminate many tools, allowing attention to be focused on the remaining candidates. Information to be obtained includes:

Vendor general information (e.g., business history, available support, plans & strategies).

Vendor's specific product development strategy.

The tool's cost (e.g., price, maintenance, modifications, training).

The hardware and software required to support tool use.

The hardware and software required to support final application/product use.

The training required for efficient tool use.

The tool's functional capabilities.

The tool's methodology and life-cycle support.

How the tool interfaces to external systems.

The number of users, existence of a user's group, the users' response to the tool.

The tool's license mechanism (e.g., floating license, multi-user licenses, cross platform licenses).

6.3 Identifying final candidate CASE tools

When the set of potential candidate tools has been identified, the final candidates for selection (those to be evaluated) may be chosen. This is accomplished through the following tasks:

Establish a set of high-priority or critical requirements to be met by CASE tools.

Compare the user's functional requirements with the CASE tool's functional capabilities, supporting methodology, system environment.

Compare the managerial requirements with the CASE tool's cost, available training and support.

Analyze the tool vendors' user base, user response, support and business history.

Identify tools satisfying a sufficient number of high-priority or critical requirements which then become the final candidates for formal evaluation. The results of the previous tasks provide the justification for the list of candidates.

NOTE - The tasks described in this paragraph represent a "screening" of possible candidates to allow the organization to identify the candidates most likely to be acceptable, given the organization's requirements or suppliers abilities. The identification of final candidates can be performed in parallel with CASE tool information gathering, or the two activities may be iterated. The goal is to reduce the cost of tool evaluation by only considering a screened set of final candidates during the evaluation process.

7 Evaluation process

Evaluation can begin when the structured requirements have been defined and a screened set of final candidates for selection have been chosen. Final preparations will be made for evaluating the candidate CASE tools, including the development of an evaluation plan. The evaluation activities are then performed and documented, resulting in a profile of how each CASE tool measures up to the structured requirements. The objective is to produce the technical evaluation reports necessary for the selection process, as shown in Figure 4.

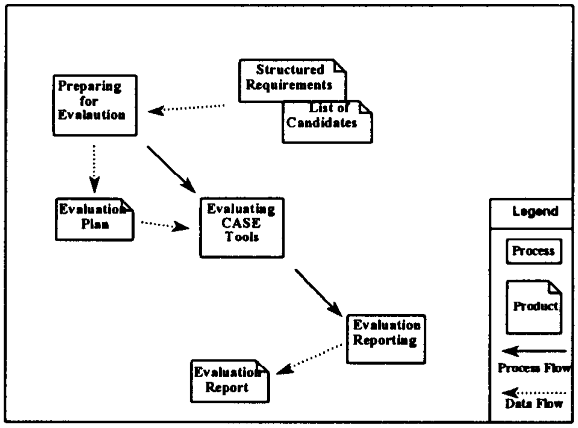

Figure 4 - Overview of evaluation process

7.1 Preparing for evaluation

To define the necessary level of detail prior to beginning evaluation activities, final preparations are necessary. Based upon the list of candidate CASE tools and the structured requirements, the following tasks should be accomplished:

For each atomic subcharacteristic, define or select one or more metrics and define the details of their use.

NOTE 1 - ISO/IEC JTC1 SC7 Working Group 6 has technical reports relating to metrics under development which may help the user of this International Standard select some of the necessary metrics.

Set the rating levels and identify the means by which the levels will be generated or computed.

NOTE 2 - A measured metric value (e.g., average lines of code per module = 274) must then be assigned a rating value (e.g., 1.3 on a scale of 0 to 4). The means by which rating levels are obtained from measurements must be identified.

Define the assessment characteristics for evaluation, establishing what is acceptable, taking into consideration the rating levels previously defined and the context of use of the product.

Identify and schedule all activities which must be performed as part of the evaluation process.

NOTE 3 - Activities include preparing any data sets necessary for the evaluation, obtaining tool documentation and an instance of the tool to be evaluated, providing evaluators any necessary training in tool use, hands-on tool use, recording of tool outputs, and analysis of results.

In some cases, a Bench Mark Test (BMT) may be a part of the evaluation process. The recommended approach for a BMT includes:

- Identify the required critical tool functions.
- Identify a test project or sample program to be the basis for the BMT.
- Develop a BMT scenario, defining inputs and expected outputs.

To focus evaluation activities and provide for traceability of the evaluation process, develop an evaluation plan which includes the information above.

7.2 Evaluating CASE tools

The software is evaluated in comparison with each of the chosen characteristics. Evaluation is a process of measurement, rating and assessment.

7.2.1 Measurement

Measurements can be made based upon information obtained by examining the CASE tool itself, or information about it, through the following types of tasks:

Examining the vendor-supplied documentation.

Examining the source code and other intermediate products, if available.

Interviewing actual users of the software.

Viewing demonstrations and interviewing demonstrators.

Executing test cases.

Applying to test projects.

Examining results of previous evaluations (whether in-house, third party, or other evaluations).

Performing a BMT on the candidate tools and analyzing the results.

Measurement values may be binary, based on a continuous scale (quantifiable), or textual. There are both objective and subjective characteristics.

NOTE - Objective characteristics are those which permit independent and repeatable test or metric. Subjective characteristics are those for which no independent and repeatable test or metric exists (e.g., fitness of the user interface to the culture of the user).

For objective characteristics, the evaluation should be made by a repeatable procedure such that another evaluator would be able to produce the same results. During evaluation, if test cases are used, a uniform, predefined, and documented set of cases should be used.

For subjective characteristics, the evaluation should be performed repeatedly by more than one person or group, who will discuss and agree upon results.

The evaluation results should be recorded in a quantified manner, where possible, together with textual justification, where applicable.

7.2.2 Rating

In the rating task, each measured value is rated against the scale of values defined in the evaluation plan. Rating levels are either directly generated or computed according to previously defined algorithms.

NOTE - It is possible that requirements may be revised during the evaluation, and this may require revision of rating scales.

7.2.3 Assessment

Based upon the resulting ratings and the previously defined assessment criteria, assess the subcharacteristics and characteristics. In accordance with the selection guidelines and the evaluation plan, ratings should be aggregated up to the characteristic level.

7.3 Evaluation reporting

The end result of the evaluation activities will be an evaluation report. An evaluation report may address all tools which were evaluated; alternatively, several evaluation reports may be written, each reporting on a subset of the tools evaluated. The evaluation report should contain at least the following information:

Tool information.

- CASE tool name
- CASE tool version
- vendor
- host configuration
- cost elements
- background, as appropriate
- part(s) of the life-cycle for which the CASE tool(s) is intended
- type of software model the CASE tool(s) is based on (e.g., waterfall model, spiral model)
- CASE tool software environment (e.g., programming language(s) supported, method supported, operating systems, possible configurations, configuration used in evaluation, minimum configuration, database compatibility, software of other vendors required for the environment)
- CASE tool functions
- input/output structure
- target audience

Evaluation process.

The report should discuss the specific activities and tasks in the evaluation process in the detail necessary to allow the reader both to understand the scope and depth of the evaluation and to repeat the evaluation, if desired.

Specific results.

Evaluation results should be provided in terms of the lowest level of subcharacteristic decomposition (normally an atomic subcharacteristic). For each subcharacteristic, the metric value measured should be given in terms of the rating level for that metric.

Based upon the lowest level results, any aggregation should be shown so as to make clear the method of aggregation: any weights used, the elements aggregated, and the level to which aggregation is performed. The result will be a profile describing the results of the evaluation in terms of scores for the characteristics of clause 9, or in terms of scores for subcharacteristics, depending on the level of aggregation.

In cases where the report covers multiple tools, or where the results of this evaluation will be compared to those of other evaluations, care should be taken to ensure the results are provided in a uniform format which facilitates comparison (e.g., by using templates). Objective results should be provided with minimal accompanying text. Subjective results should be supported by text describing the specific reasons for the metric values assigned.

NOTE - The information specified above could be organized as follows:

Evaluation process
Goals, criteria, tools evaluated
Measurement tools
Tool information
Test scenario
Test results and evaluation
Evaluation summary

8 Selection process

Selection can begin when the evaluation reports are complete. A selection algorithm should be defined and then applied to the results of the CASE tool evaluation efforts. A decision can then be recommended, and the recommendation validated against the original set of goals and selection guidelines. An overview of the process is shown in Figure 5.

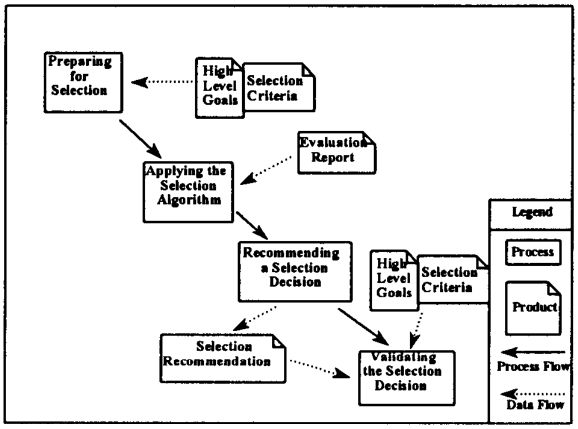

Figure 5 - Overview of selection process

8.1 Preparing for selection

The selection algorithm determines how the data generated during the various evaluations are combined and compared to result in ratings for each candidate.

Based upon the original goals and selection guidelines, a final set of selection criteria is identified and the basis upon which these criteria are to be assessed is defined. This definition will be based upon the aggregated evaluation assessments described in 7.2.3.

The algorithm for further aggregating the results, comparing the candidates, and arriving at a decision is then defined. A discussion of selection algorithms is provided in Annex B.

8.2 Applying the selection algorithm

The evaluation results are used as inputs to the selection algorithm. Information relating the candidate tools is output. Each tool's evaluation results provide a technical summary of each tool's characteristics, aggregated up to the level specified in the selection algorithm (usually the characteristic level). The selection algorithm combines the results of the evaluations of the various tools, providing a comparison for use by decision makers.

8.3 Recommending a selection decision

When the selection algorithm has been applied, a decision may be made to acquire a tool or set of tools. This is a management decision based upon the technical comparison provided above and additional management criteria.

Such a decision would indicate that the most appropriate of the candidates has been identified for selection. Alternatively, the assessment of evaluation results may show a need for additional information, which may indicate that some iteration of previous activities is necessary. Evaluation and selection scenarios are further discussed in Annex A.

The selection decision should be justified with a rationale which summarizes the information and logic which led to the selection.

8.4 Validating the selection decision

The final activity in the process should be the validation of the recommended selection. The original goals and selection guidelines should be reviewed and compared to the evaluation results and other data relating to the recommended selection. A check should be made to ensure that if the recommendation is accepted, the high level goals (or a sufficient number of them) will be met.

It may be found that no adequate tool exists, in which case a choice may be made between the development of a new tool or the modification of an existing tool (within the user organization or outside), or abandoning the entire evaluation and selection process.

9 CASE tool characteristics

The user needs which drive any evaluation and/or selection process will be based upon the characteristics and subcharacteristics described below. By defining user needs in the terms used here, assessments and comparisons may be made based upon a broad, common, and nearly complete set of characteristics. As discussed above, a structuring activity is required to transform the set of needs initially identified by the user into the terms provided here.

The top-level evaluation categories are called characteristics. Each characteristic is subdivided into subcharacteristics. Subcharacteristics may be further subdivided into lower level subcharacteristics. At the lowest-level, subcharacteristics are referred to as atomic subcharacteristics. This section defines atomic subcharacteristics in terms of their attributes; each of these will be assigned a value during the evaluation process based upon one or more metrics (see 7.1, Preparation for evaluation).

It is unlikely that any user of this International Standard will need to use all of the atomic subcharacteristics given below; users should select only those subcharacteristics which have significant weight with respect to their organization's requirements. There will be cases where additional needs or characteristics, specific to a particular evaluation or selection, have to be added to those listed below; in that sense, the atomic subcharacteristics listed below can be considered a partial list, to be augmented as necessary.

Non-atomic subcharacteristics are assigned values by aggregating the values of their component subcharacteristics, weighted as called for in the evaluation plan. This aggregation task is continued until the levels of aggregation called for in the evaluation plan have been reached. The selection algorithm is then used to combine the evaluation results of the various tools for comparison and decision.

9.1 Functionality - characteristics related to life-cycle processes.

A set of attributes that bear on the existence of a set of functions and their specified properties to support CASE tool use as it relates to software engineering life-cycle processes and activities. For those life-cycle processes referenced, the definitions in ISO/IEC 12207:1995 apply.

NOTE - This section addresses CASE support for several life-cycle processes. Other life-cycle processes not addressed here are absent either because CASE tools do not typically provide support for those processes, or because the process and/or the CASE support for it are not stable at this time.

9.1.1 Characteristic: Management Process

A set of attributes that bear on the existence of a set of functions and their specified properties to support the management process activities. For additional attributes that bear on management, see 9.1.6 for risk management.

Atomic subcharacteristics:

Cost and Schedule Estimating: attributes relating to its ability to estimate cost, schedule and other project parameters based upon organizational inputs.

NOTE 1 - For example, the Constructive Cost Model (COCOMO) and its variants.

Planning: attributes relating to its ability to support user entry and analysis of project planning data.

NOTE 2 - This subcharacteristic is more general than the subcharacteristic above; in addition to cost and schedule data, it includes, for example, computer and other facility resources, personnel allocations, annual calendar definiton and vacation planning. Also included is the capability of analysis of planning data, such as a critical path analysis to optimize the project plan with respect to the required constraints, and the capability of reusing/modifying the planning data.

Project Tracking: attributes relating to its ability to support user entry of project activity data, including automated data gathering.

NOTE 3 - Examples of project activity data which may be tracked include completion date, funds expended, resources consumed, number of documents generated, lines of code developed, number of test cases completed, and number of defects detected.

Project Status Analysis and Reporting: attributes relating to its ability to support analysis of project activities based on the data tracked and provide status reports and projections in user definable formats.

Managing Processes: attributes relating to its ability to support the management of processes.

NOTE 4 - Process management includes defining detailed work items by defining input, resources, output, personnel, deadline, etc.; making work item definitions available to project members; updating work status by (manually or automatically) gathering the results of the work. Query capability is also included.

9.1.2 Characteristic: Development Process

A set of attributes that bear on the existence of a set of functions and their specified properties to support the development process activities. For additional attributes that bear on the development process, see 9.1.8.

NOTE - The set of development functions may not be exhaustive, and additional subcharacteristics may be considered as required.

9.1.2.1 Subcharacteristic: Modeling

A set of attributes that bear on the existence of a set of functions and their specified properties to support the modeling activities which can be part of the development process.

NOTE 1 - Modeling functions reflect the CASE tool's ability to support the identification of software requirements, the expression of software design, and the transformation of requirements into design.

Atomic subcharacteristics:

Diagram Development: attributes relating to its ability to support the entry and editing of diagrams of types of interest to the user, and to translate between diagram types, and between diagrams and text.

NOTES

2 - Diagram types are defined in ISO 5807:1985. In addition, typical diagram types include: control flow, data flow, decomposition, entity-relationship, object oriented, Petri nets, state transition, and structure charts.

3 - Rules relevant to specific diagram types should be enforced.

Diagram Analysis: attributes relating to its ability to support the analysis of graphical figures input to the CASE tool and extracting and storing requirements and/or design information.

NOTE 4 - Diagram analyzers are, in many cases, integrated with diagram drawers, but go beyond diagram drawers in analytical capability.

Requirement Specification Support: attributes relating to its ability to support the entry and editing of requirements specification data and checking the consistency and completeness of the requirements data against allowable specification constructs and rules.

NOTES

5 - Classes of requirements data which may be considered include: functional, data, interface, quality, performance, hardware, environment, cost, and schedule requirements.

6 - A formal language may be used to express requirements data.

Design Specification Support: attributes relating to its ability to support the entry and editing of design specification data and checking the consistency and completeness of design data against allowable specification constructs and rules.

NOTES

7 - Classes of design data which may be considered include: functional, data, interface, quality, performance, hardware, environment, cost, and schedule information.

8 - A formal language may be used to express requirements data.

Specification Construct Modeling: attributes relating to its ability to support the entry and editing of information describing the types of constructs that a specification can contain, including their relationships and depiction.

NOTE 9 - Types of constructs which might be modeled include data structures, data flows, objects, processes and states.

Simulation: attributes relating to its ability to simulate aspects of a system's potential operation based upon requirements and/or design data available to the CASE tool.

NOTE 10 - Examples of aspects to be simulated include system effectiveness (operational utility), operator interface, and architectural performance (response time, utilization, throughput).

Prototyping: attributes relating to its ability to generate a prototype model of all or portions of a system based upon user-supplied requirements and/or design information.

NOTE 11 - Prototyping features of CASE products may sometimes be replaced by 4GL/graphical user interface (GUI) tools. Such replacement requires fluent transition from modeling to design activities and back.

Human Interface Modeling: attributes relating to its ability to model the content aspects of human-computer interactions and the mechanical aspects of those interactions.

NOTE 12 - Examples of content aspects of human-computer interactions are presentations (e.g., menus, screen and window layouts and report designs) and querying (e.g., text, voice, touch, icon or other inputs). Examples of mechanical aspects include window location, size, and colors; voice volume and pitch, and touch sensitivity).

9.1.2.2 Subcharacteristic: Construction

A set of attributes that bear on the existence of a set of functions and their specified properties to support the construction activities which can be part of the development process.

NOTE 1 - Construction functions reflect the tool's ability to produce operational (e.g., executable) elements of the final system to be fielded, or to modify an existing system. Many of the functions in this paragraph are dependent upon a specific language or languages. Examples of such languages include programming languages, data and query languages, graphics languages, and operating system interfaces such as job control languages. The user of this International Standard should identify those languages relevant to the specific effort.

Atomic subcharacteristics:

Code Generation: attributes relating to its ability to generate code in one or more specific languages based upon design data available to the CASE tool.

NOTE 2 - Typical code generation capabilities include general purpose code generation, database generation, query generation, screen display/menu generation. Another form of code generation is the direct generation of executable code.

Database Schema Generation: attributes relating to its ability to generate database schema based upon user-supplied information.

Screen Generation: attributes relating to its ability to generate display screens based upon user-supplied information.

Report Generation: attributes relating to its ability to automate the development of reports to be produced by the system under development (as opposed to the CASE tool).

Compilation: attributes relating to its ability to compile code in one or more specific languages.

Syntax Directed Editing: attributes relating to its ability to support the entry of source code in one or more specific languages with syntax support provided by the editor.

Debugging: attributes relating to its ability to support the identification and isolation of errors in a program.

NOTE 3 - Typical capabilities include providing tracebacks and identifying fault location in terms of source code.

9.1.3 Characteristic: Maintenance Process

A set of attributes that bear on the existence of a set of functions and their specified properties to support the maintenance process activities.

Atomic subcharacteristics:

Problem understanding: attributes relating to its ability to determine that a problem: results from a user misunderstanding, has already been resolved, is going to be resolved in the context of another maintenance action, or is a new problem to be resolved.

Localization: attributes relating to its ability to identify the portion of the software requiring modification, given the identification of a problem.

Impact analysis: attributes relating to its ability to, for each change foreseen, identify potential consequences of making the change.

Data Reverse Engineering: attributes relating to its ability to extract information from source code which defines or describes the data elements and structures of the software.

NOTE 1 - Examples of typical outputs include design language code, data dictionary entries, and direct entry of design data into the CASE tool's database.

Process/Procedure Reverse Engineering: attributes relating to its ability to extract process design data from source code.

NOTE 2 - Examples of typical outputs include design language code, design diagrams, and direct entry of design data into the CASE tool's database.

Source Code Restructuring: attributes relating to its ability to input existing source code in one or more specific languages, modify its format and/or structure according to defined directives (e.g., reduce size of code, reduce execution time, implement code format standard) and output a source code file in the same language.

NOTE 3 - Examples of typical capabilities are pretty printers and source-level optimizers.

Source Code Translation: attributes relating to its ability to input existing source code written in one or more specific languages, translate it into a different language, and output the resulting code.

9.1.4 Characteristic: Documentation Process

A set of attributes that bear on the existence of a set of functions and their specified properties to support the documentation process activities.

Atomic subcharacteristics:

Text Editing: attributes relating to its ability to edit text.

Graphical Editing: attributes relating to its ability to enter and edit data in graphical format.

Forms-Based Editing: attributes relating to its ability to support user definition of forms and subsequent forms-based editing.

Publishing: attributes relating to its ability to support desktop publishing.

Hypertext Support: attributes relating to its ability to support hypertext formats and functions.

Variant Handling: attributes relating to its ability to reuse the same generation of the product with limited variation.

NOTE - Examples include change of objects in screen panes (e.g., logo) and adaptation to local requirements (e.g., language).

Automatic Data Extraction and Document Generation: attributes relating to its ability to accept, store and retrieve specifications of the content, format and layout of textual and graphical data to be extracted and produced, and its ability to then extract and produce the data in compliance with a specification.

9.1.5 Characteristic: Configuration Management Process

A set of attributes that bear on the existence of a set of functions and their specified properties to support the configuration management process activities.

Atomic subcharacteristics:

Access Control: attributes relating to its ability to control access to data elements.

NOTE 1 - Access control includes the ability to specify components to be no access, read only, etc. based upon work groups, or other similar identifier, as well as the ability to check-out data elements for modification and restrict access to them (locking) until they have been updated and checked back in (unlocking).

Tracking of Modifications: attributes relating to its ability to maintain a record of all modifications made to the system under development or maintenance.

NOTE 2- As design and code information is changed, includes automatically updating and keeping consistent all related information kept in the tool.

Definition and Management of Multiple Versions: attributes relating to its ability to maintain records and perform management functions on multiple versions of a system which may share common components.

Configuration Status Accounting: attributes relating to its ability to provide the user with reports defining the history, contents and status of the various configuration items being managed.

Release Generation: attributes relating to its ability to support user definition of steps required to create a version (build) of the software for release, and to automatically execute those steps.

Archival Capability: attributes relating to its ability to automatically place data elements in secondary storage for subsequent retrieval.

NOTE 3 - Archiving typically involves long term storage of information off-line for use in reconstructing a system which was damaged or accessing data which is seldom needed. Archiving features which may be considered include level of automation, ease of retrieval, data compression capabilities and security against both loss and unauthorized access.

9.1.6 Characteristic: Quality Assurance Process

A set of attributes that bear on the existence of a set of functions and their specified properties to support the quality assurance process activities.

Atomic Subcharacteristics:

Quality Data Management: attributes relating to its ability to support user entry of quality data, analyze quality data, and generate information to support quality management.

NOTES

1 - Examples of quality data include quality assurance plans, results of reviews and audits, test

results, fault and corrective action reports, and values of complexity metrics.

2 - Includes the ability to handle quality data by variant.

Risk Management: attributes relating to its ability to support risk identification, risk estimation, risk impact assessment, risk monitoring and controlling.

NOTES

3 - Risks to be analyzed might be categorized into, but not limited to, project risks, technical risks, and business risks.

4 - Risk management support capabilities might include hazard analysis in critical applications such as air traffic control systems or nuclear power plant control systems.

9.1.7 Characteristic: Verification Process

A set of attributes that bear on the existence of a set of functions and their specified properties to support the verification process activities. For additional attributes that bear on the verification process, see 9.1.8; additional functional capabilities relating to verification may be found embedded in existing development subcharacteristics.

Atomic subcharacteristics:

Specification Traceability Analysis: attributes relating to its ability to perform traceability analyses.

NOTE 1 - Analyses normally address information from the level of requirements specifications through design data.

Specification Analyses: attributes relating to its ability to perform analyses based upon the requirements and design data available to the tool.

NOTE 2 - Specific types of analyses might include: algorithm, complexity, control flow, data flow, data normalization, data use, interface, human-machine interface, range bound, and structure.

Source Code Analysis: attributes relating to its ability to input source code in one or more specific languages and perform analyses.

NOTE 3 - Examples of such analyses include the measurements of size, calculation of complexity metrics, generation of cross-references, and review for conformance to standard usages.

9.1.8 Characteristic: Validation Process

A set of attributes that bear on the existence of a set of functions and their specified properties to support the validation process activities.

Atomic subcharacteristics:

Proof of Correctness Techniques: attributes relating to its ability to formally prove assertions about features or operations of the software to be validated.

Failure Analysis: attributes relating to its ability to analyze failures and trace them back to defects.

Defect Analysis: attributes relating to its ability to analyze defects and trace them forward to failures.

Test Case and Expected Result Entry: attributes relating to its ability to support user entry of test cases and entry of expected test case results.

Test Case and Expected Result Generation: attributes relating to its ability to automatically generate test cases based upon existing requirements and/or design specification data available to the tool and to automatically generate expected test case results.

Test Traceability: attributes relating to traceability of test activities and data.

NOTE 1 - Aspects include traceability of test data to other test data (e.g., test requirements to test design to test cases) as well as traceability of test data to activities and data from other life-cycle activities (e.g., requirements specifications to test cases and test cases to source code).

Source Code Instrumentation: attributes relating to its ability to automatically instrument code to be tested in order that test events can be identified and recorded.

Input Capture and Replay: attributes relating to its ability to capture operator inputs (e.g., keyboard, mouse) and the extent to which such data can be edited and replayed in subsequent test cases.

Test Driving: attributes relating to its ability to execute and/or replay test cases.

Run-time Analysis: attributes relating to its ability to analyze the performance of a program as it executes.

NOTE 2 - Capabilities may include the ability to verify and report on assertions (or exceptions) encountered during execution, as well as reporting on CPU utilization, memory utilization, accesses to specified data elements and/or code segments, and timing characteristics.

Reliability Analysis: attributes relating to its ability to analyze measures of software reliability.

NOTE 3 - Examples of reliability measures include measures of complexity, software science attributes and the mean time between failure (MTBF).

Test Coverage Analysis: attributes relating to its ability to analyze and report on test coverage, including system coverage analysis and function coverage analysis.

NOTE 4 - For example, statements which were/were not executed, procedures which were/were not called, and variables which were/were not accessed.

Test Procedure Management: attributes relating to its ability to manage test activities and a test program.

NOTE 5 - For example, the ability to maintain a schedule of planned activities, capture and record the results of test activities, and generate status reports.

Regression Testing: attributes relating to its ability to support regression testing.

NOTE 6 - For example, the ability to re-run previous tests; the ability to modify previous tests to account for system and/or environmental differences (e.g., date, time).

Automatic Result Checking: attributes relating to its ability to automatically compare expected test case results and actual test case results.

Test Statistical Analysis: attributes relating to its ability to statistically analyze and report on test results.

NOTE 7 - For example, percent of test cases executed and per cent of test cases passed.

Operations Environment Simulation: attributes relating to its ability to support the simulation of a real operations environment, such as a large number of users, as well as various scenarios of use and various configurations.

NOTE 8 - For example, the ability to automatically generate simulated inputs to the system being tested based upon received system outputs.

Integration Testing: attributes relating to its ability to support software integration activities.

NOTE 9 - For example, the automatic generation of body stubs for top-down testing or the automatic generation of driver procedures for bottom-up testing.

9.2 Functionality - characteristics related to CASE tool usage.

The following characteristics relate the tool to its environment and the projects it will be used to support.

9.2.1 Characteristic: Environment in which the CASE tool operates.

A set of attributes which bear on the relationship between the CASE tool and its operational (host) environment.

Atomic subcharacteristics:

Required Hardware Characteristics of Tool: attributes relating to any hardware requirements for its use.

NOTES

1 - Typical hardware items to be listed include processors (including co-processors), main memory size, bus type, type and size of peripheral storage, extension or graphics cards, input and output equipment.

2 - The user of this International Standard should identify hardware items for which the minimal requirements may not provide adequate performance, e.g., main memory. Hardware necessary to provide acceptable performance should be identified.

3 - The user of this International Standard should identify hardware items which are supported by the tool as options, e.g., input and output devices.

Required Software Environment of Tool: attributes relating to any software items required for its use.

NOTE 4 - Typical software items to be listed include operating systems, database management systems, languages, character sets and character codes, and communications/network packages.

Software Repository (Information Base): attributes relating to its ability to house and manage all relevant software engineering process information. This includes its ability to make information developed in one life cycle activity available for use during other activities, as well as its ability to provide access to this information to other environment elements.

NOTES

5 - Examples of such information include requirements and design documentation, code, and test data.

6 - Includes the ability to handle relevant data by variant.

Physical Environment of Tool: attributes relating to any geographical aspects of the development environment which will impact tool use.

NOTE 7 - Considerations include physical and temporal separation of users and the related issues of networking considerations, on-line/off-line considerations, and repository updating/mirroring at multiple sites.

9.2.2 Characteristic: CASE tool integrability.

A set of attributes which bear on the ability of the CASE tool to integrate and interoperate with other items in its operational environment. The evaluation and selection of CASE tools is performed in the context of the software engineering environment in which the tool will be used.

NOTE - Examples of other environmental items include those given in the hardware and software environment of the tool, above.

Atomic subcharacteristics:

Compatibility with Environment Elements: attributes relating to its ability to interoperate with and/or directly exchange data with hardware/software environments.

NOTES

1 - Examples of other software tools include word processors and other documentation tools, databases, repositories, and other CASE tools.

2 - If the tool contains an interfacing capability (e.g., an application programming interface) which allows the tool to be used independently of environment elements, that interface should be described.

3 - The extent to which the tool conforms to standards for "openness" including data interchange formats, can be evaluated in terms of a number of existing standards, including, for example, ECMA TR 55, *A Reference Model for Frameworks of Software Engineering Environments*, ECMA 149, *Portable Common Tool Environment (PCTE) Abstract Specification*, ISO/IEC 9945-1: *POSIX* and ISO 9075-*SQL*.

4 - Process management, project management and system development functions may be provided by separate, specialized tools. In such a case, the connectivity between the different tools should be considered under this subcharacteristic.

Data Integration: attributes relating to its ability to use, process, and deliver information shared by other tools or part of a repository.

NOTE 5 - The CASE tool should be evaluated against:

Metadata: if the framework is based on a specific data model (e.g., Entity/Relationship or Object Oriented), the ability of the CASE tool to generate, manipulate or access compatible structures of the relevant data model (e.g., compatible type, constraint, attribute or relationship).

Product data engineering: if the repository holds a formal data description for software engineering, the ability of the CASE tool to generate, manipulate or access relevant data according to its functional scope.

Control Integration: attributes relating to its ability to interact with the SEE, in particular, with other tools.

NOTES

6 - Invocation of the tool may be controlled by rules, e.g., security policies, pre- or post-invocation of other tools, allowable concurrency and synchronization. The rules may be defined by the supported method. The level of compatibility of rules might be addressed.

7- Different communication mechanisms exist in different frameworks. Communication between tools can be handled by sharing data within a repository, by message queues between tools, by client-server approaches, or by remote procedure calls. For example, consider the X3.46 control integration and Portable Common Tool Environment (PCTE) approaches..

Presentation Integration: attributes relating to the level of the homogeneity, compatibility, and consistency of its user interface with that of the remainder of the SEE.

NOTES

8 - With respect to all integration characteristics, if the CASE tool was not developed specifically for the framework, attention should be addressed on ways of adapting it to the framework (e.g., encapsulation). For example, an important issue is the user interface, which might not be homogenous with those of other tools despite the fact that the tools share a common framework.

9 - In the case that the framework is defined by a specification (e.g., expressed in a language such as C or Ada), the specification of the tool interface should be comparable.

Metadata Access: attributes relating to the access provided to the tool's metadata.

9.2.3 Characteristic: Aspects of the CASE tool's application.

A set of attributes which bear on the relationship between the CASE tool and the projects to which it is applied, including the environment of its products and characteristics of those products.

Atomic subcharacteristics:

Hardware and Software Environment of Tool Products: attributes relating to the set of hardware and software items on which or with which the products of the tool can be used.

NOTE 1 - The level of platform support in the target environment may be a consideration; e.g., does the CASE tool generate screens, and does it generate calls to an external (platform or environment) service to generate the screens.

Conformance to Standards - Tool Products: attributes relating to conformance of the products resulting from its use to standards.

NOTE 2 - Examples include language, database, repository, communication, GUI, documentation, development, configuration management, security, portability, and information interchange standards.

Domain of Application: attributes relating to the application domains which the CASE tool is designed to support.

NOTE 3 - Examples of application domains include transaction processing, real-time, information management, and safety critical.

Size of Application Supported: attributes which would result in size limitations of the application and therefore limit the tool's applicability.

NOTE 4 - Such parameters might include lines of code, levels of nesting, size of database, number of data elements, and number of configuration items.

Languages Supported: attributes relating to its ability to support specific languages.

NOTE 5 - Examples of such languages include programming languages, data and query languages, graphics languages, and operating system interfaces such as job control languages.

Databases Supported: attributes relating to its ability to support specific databases.

Methodology Support: attributes relating to the set of methods or methodologies which it can support.

NOTES

6 - Examples of methods or methodologies include various object oriented approaches, structured (top down) approaches, data driven approaches, and real-time extensions.

7 - A CASE tool's support for a method or methodology can be evaluated based upon the extent that the tool provides the specific capabilities necessary to implement the methodology.

Internationalization: attributes relating to its ability to be used in different cultures and to generate products in terms of different countries or cultures.

NOTES

8 - Examples include using natural different languages, character sets, character and graphic presentation modes (left-right, top-bottom), and different date formats.

9 - This subcharacteristicmay have an influence on other hardware or software environment elements.

9.3 General quality characteristics

The following characteristics describe the quality of the tool in the terms of ISO/IEC 9126.

NOTE - Further guidance on evaluating the subcharacteristics in this section can be found in ISO/IEC 12119-1994.

9.3.1 Characteristic: Functionality

A set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those which satisfy stated or implied needs.

Atomic subcharacteristics:

Security: attributes relating to its ability to prevent unauthorized use or misuse of itself.

NOTE 1 - Security may also encompass all or part of the system on which the tool is used.

Accuracy: attributes relating to the provision of right or agreed results or effects.

Regulatory Compliance: attributes relating to adherence to application related legal and/or regulatory requirements.

Technical Compliance: attributes relating to adherence or compliance with any identified standards.

NOTE 2 - Examples include language, database, repository, communication, GUI, documentation, development, configuration management, security, portability, and information interchange standards.

9.3.2 Characteristic: Reliability

A set of attributes that bear on the capability of the software to maintain its level of performance under stated conditions for a stated period of time.

Atomic subcharacteristics:

Data Integrity: attributes relating to its ability to correctly store and retrieve information with a high degree of confidence.

Automatic Backup: attributes relating to its ability to automatically initiate a backup routine to save the current state of the process.

NOTE 1 - Typically backups are scheduled at a predetermined interval by the vendor or are scheduled by the user.

Error Handling: attributes relating to its ability to detect abnormal behavior, notify the user that a problem has occurred, and properly exit or save the work to the point of interruption.

NOTE 2 - This may include error messages displayed to the screen and a screen directed means of either exiting or saving.

Fault Tolerance: attributes relating to its ability to maintain a specified level of performance (e.g., “failsoft” or reduced capability) in cases of various faults (e.g., hardware, software, network).

Recoverability: attributes relating to its capability to re-establish its level of performance and recover the data directly affected in case of a failure, and the time and effort needed for it.

9.3.3 Characteristic: Usability

A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.

Atomic subcharacteristics:

User Friendliness: attributes relating to its ability to integrate into the tool user's activities, taking into account the user's level of experience and expertise, and the concepts, information, representations and procedures that are part of the user work domain and culture (professional and individual).

User Guidance: attributes relating to the provisions to allow the tool user to know the status of tool operation, to establish the causal relationship between user actions and tool status, and to assess and direct user actions on the tool.

NOTE 1 - Capabilities may be provided in the form of on-line help features, and diagnostic and error messages.

Homogeneity: attributes relating to the consistency of logic within an application or across applications, at the procedural level as well as for the presentation of information.

Adaptability: attributes relating to the ability of its interface to adapt to various task requirements, strategies, habits, and cultural modes (e.g., languages, character sets, date formats).

NOTE 2 - Adaptability has several aspects: the ability to adapt to users with differing levels of experience, the ability of the user to customize input and output methods (e.g., macros and screen displays and formats), and in the number of procedures, options and commands available to a user to achieve a given objective.

Clarity of Control: attributes relating to the extent to which the semantics of the dialogue steps (e.g.,menu selections, window choices) used to control the tool, reflect the resulting action, and the predictability of the action.

Error Handling: attributes relating to its abilities to help and guide the user in identifying and correcting errors, and to maintain tool integrity (avoiding incorrect data and process changes).

Conciseness: attributes which decrease the required number of steps to identify and memorize, and which increase the efficiency of the dialogue.

Ease of Learning: attributes relating to the amount of time and effort required for a user to understand normal CASE tool operations and to become productive.

NOTES

3 - The availability and quality of on-line tutorials may be a factor in ease of learning.

4 - These features should be integrated within the presentation and structure of the data on the screen (or reports).

Tool Documentation Quality: attributes relating to the overall quality of the documentation provided with the tool.

NOTES

5 - Documentation quality factors include: completeness, correctness, consistency, understandability, and ease of overview.

6 - To the extent that the tool implements a methodology, descriptions of the methodology should accompany the tool.

Ease of Installation: attributes relating to how easy it is for the user to play the required role in initial installation and in the subsequent installation of updates.

9.3.4 Characteristic: Efficiency

A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions.

NOTE - In evaluating a CASE tool against the subcharacteristics which follow, consideration should be given to jobs of both typical and maximal size.

Atomic subcharacteristics:

Throughput: attributes relating to the performance of the tool in performing stated tasks.

NOTE 1 - Examples include query response time and time to analyze 100,000 lines of code. In some cases, performance benchmark data are available from external sources.

Acceptable Response Time: attributes relating to the acceptability of the time required for the CASE tool to respond to a user input with the appropriate response in the expected operational environment.

Data Storage Requirements: attributes relating to the amount of mass storage required (e.g., disk, tape) to accommodate the tool itself and any databases required and/or generated by the tool.

Acceptable Memory Capacity: attributes relating to the amount of CPU addressable memory required to load and operate the tool.

NOTE 2 - A determination of the amount of memory required for acceptable tool performance should be made, as many tools will operate in a memory-poor environment, but will do so only marginally.

Acceptable Processing Speed: attributes relating to the processor (type and speed) required to operate the CASE tool at an acceptable level of performance.

9.3.5 Characteristic: Maintainability

A set of attributes that bear on the effort needed to make specified modifications.

Atomic subcharacteristics:

Vendor Support: attributes relating to the availability, responsiveness, and quality of services provided by the vendor to tool users.

NOTE 1 - Such support services might include telephone support, local technical support, on-site support, publication of "known defect lists".

Ability of Tool to Follow Changes in Methodology: attributes relating to the ability of the tool vendor to modify the tool to maintain methodology support, as a methodology changes over time.

Updates: attributes relating to the vendor's track record in making regular updates which address recognized problems and/or provide additional capabilities.

Expandability: attributes relating to its ability to be easily modified to meet expanded user needs without requiring major modification, expense, or schedule change.

NOTE 2 - Related to changeability, except here, vendor intervention or additional hardware and/or software may be required.

9.3.6 Characteristic: Portability

A set of attributes that bear on the ability of the software to be transferred from one environment to another.

Atomic subcharacteristics:

Portability to Different Hardware Platforms: attributes relating to its ability to run on various versions of the same hardware platform or on different hardware platforms.

Compatibility With Different Operating Systems: attributes relating to its ability to run on various versions of the same operating system or on different operating systems, and the ease with which it can be modified to run on updates to an operating system.

Ability to Move Data Between Versions of the Tool: attributes relating to the ability of one version of the tool to use data generated by a different version of the tool, and the extent of data manipulation required for reuse.

Portability with Windowing Systems: attributes relating to its ability to be ported between windowing systems (e.g., Open Look and Motif).

9.4 General characteristics not related to quality

The following characteristics are general in nature, and address both the tool itself, the tool developer and/or vendor:

9.4.1 Characteristic: Acquisition Process

A set of attributes that bear on the acquisition process necessary if the CASE tool is selected for adoption.

Atomic subcharacteristics:

Cost of Tool Implementation: attributes relating to the cost of implementing the tool.

NOTE 1 - All aspects relevant to the specific instance should be considered. Not only tool purchase price, but also the costs for installation, initial maintenance, hardware/software revision or upgrades, and training. Price data on all relevant configurations should be considered, including single copy, multiple copies, site license, corporate license, and network license.

Licensing Policies: attributes relating to the supplier's licensing policies.

NOTE 2 - These include available license options, the right to copy (media and documentation), and any restrictions and/or fees for secondary usage. That is, the tool user sells products which include some element or aspect of a tool used to develop the product. Also, any terms and conditions, including product guarantee, which apply to the tool.

Export Restrictions: attributes relating to the identification of any restrictions on the export of the tool, or of any secondary usage of the tool.

9.4.2 Characteristic: Implementation

A set of attributes that bear on the tool's delivery, installation and operation.

Atomic subcharacteristics:

Cost Effectiveness: attributes relating to the cost of tool operation.

NOTE 1 - A cost/benefit analysis may be performed, or some consideration given to the expected level of productivity of the CASE tool.

Development/Delivery Constraints: attributes relating to any schedule constraints involving further product development and/or delivery. In addition, the time required for the tool's users to become productive with the tool (learning curve) should also be considered.

Workarounds Required for User Organization: attributes relating to any workarounds which would be required to implement the CASE tool in the user's environment. An example of such a workaround is finding a way to use a centralized tool (single common database) in a distributed environment.

Infrastructure needs: attributes relating to infrastructure requirements for tool use.

NOTE 2 - Examples include floor space, table space, furniture, electricity and other physical requirements generated by new tool-related hardware or tool use considerations.

9.4.3 Characteristic: Support Indicators

A set of attributes that bear on the vendor's ability to provide tool support.

Atomic subcharacteristics:

Supplier Profile: attributes relating to a general indication of the supplier's overall capability.

NOTE 1 - Examples include: the supplier's size, number of years in business, market share, a financial statement, listing of any complementary products, identification of relevant business relationships (e.g., other tool suppliers), local presence, and the company's planned direction for future development.

Product Profile: attributes relating to general product use information.

NOTE 2 - Examples include: product age, number of paid installations, number of distributors of the tool, existence, size and level of activity of a user's group, number of versions supported, availability of dial up help hot-line, availability of maintenance contracts, formal problem reporting system, product development program, lead times (e.g., new functions, trouble reports, customer support), body of applications, freedom from error, and availability (commercial, government, public domain, in-house, or under development).

Training Availability: attributes relating to the availability of training materials and training courses, both at the vendor's facility and at the purchaser's facility.

NOTE 3 - Conditions under which training can be supplied, including the availability of courses customized for specific user needs should be considered.

9.4.4 Characteristic: Evaluation or Certification

A set of attributes that bear on the evaluation or certification of the developer or the product.

Atomic subcharacteristics:

Developer Evaluation or Certification: attributes relating to the evaluation or certification by a professionally recognized software engineering evaluation organization that the developer's software engineering practices meet some minimum level, or the vendor's intention to obtain such an evaluation or certification.

NOTE - For example, a capability maturity assessment based upon the Software Engineering Institute's Capability Maturity Model, or ISO 9001 certification.

Product Certification: attributes relating to the certification by an appropriate party that the tool complies with a specific standard.

Annex A
(informative)
Considerations on the use of this International Standard

There are a number of possible scenarios when evaluating and/or selecting CASE tools. Different business goals may be handled by the processes defined herein. The initiation process defined in the International Standard can be used in all scenarios to help guide the overall project plan.

In planning the work, the following two areas should be considered:

A.1 What is the scope of the CASE tool search?

The set of evaluation and selection processes may be performed to satisfy one of several objectives. These include the following:

The organization wishes to decide whether or not to purchase a specific tool. There is only one candidate; the candidate should be evaluated to determine whether its benefit will justify its purchase, and a selection decision made.

The organization wishes to provide automated support for some aspect (e.g., life-cycle phase) of its software development process. For example, it may decide it wants to provide a tool for maintaining requirements and top level design information, to trace requirements to design, and to generate related documentation. A detailed set of requirements should be defined and candidates identified, a selection algorithm adopted, and the candidates evaluated.

An organization wants to improve its software development process, but is not sure where to start. The organization should review the activities and tasks in the initiation process. Before considering the applicability of a specific CASE tool, the organization should consider a more complete assessment of its current processes. This will assist the organization in its decision whether to install a CASE tool.

A.2 To what extent can existing evaluations be used?

The user of this International Standard provides consistency in the evaluation and selection processes. This can be used to benefit those instances where an organization uses one or more evaluations that have been done at some earlier time, either by itself or by another organization. In such a case, care must be taken to ensure that the evaluations performed by different organizations are appropriate for comparison. User needs, upon which the various evaluations were based, should be compared and consideration given to the expertise, motivation, and possible biases of the organization(s) which originally performed the evaluation. In addition, the metrics used to evaluate the subcharacteristics should be examined for applicability. It should be noted that if new evaluation results are to be compared to results of previous evaluations, the metrics and rating scales used should be as similar as possible.

In another scenario, one or more CASE tools are only evaluated without proceeding to selection. For example, the evaluation(s) may be performed as a self-evaluation by a CASE tool supplier, or for entry into a data repository by a tool evaluation organization. In this case, the evaluations should be general in nature, performed against all relevant user needs, in as much as the needs of interest to future selectors cannot be determined.

A.3 Other considerations

During the evaluation and selection processes, there are other considerations to be taken into account. For example:

Information may be obtained during evaluation activities which leads to a modification of the requirements and, possibly, re-evaluation of some candidates.

Upon completion of evaluations, there may be no significant difference between the top candidates. This may be addressed by selecting arbitrarily, by modifying the selection algorithm, or by modifying the requirements and/or metrics and performing additional evaluation activities.

Levels of evaluation may be performed, interspersed with selection (or elimination) activities. For example, if a large number of candidates is identified, an initial, top-level evaluation may be performed of all candidates and a cost/benefits analysis performed to allow the elimination of some candidates. Further, more detailed, evaluation of the remaining candidates may be performed, followed by the elimination of some. This process may be repeated several times until a final selection is made.

Annex B
(informative)
Examples of selection algorithms

An important part of the selection process is the application of some algorithm to the evaluation results. Algorithms used by organizations vary widely. Selection processes rely upon some assignments of weights to characteristics, and then combining the resulting weighted evaluation ratings using some algorithm.

B.1 Considerations in assigning weights

The weights assigned to subcharacteristics and characteristics must reflect the organization's actual needs. If the assignment of weights reflects the opinion of a few individuals' individual preferences rather than the organization's actual needs, there is a higher risk that the CASE tool selected will not be successfully adopted.

To allow the evaluation to provide useful information for the selection process, the weights must provide for the discrimination between candidate tools where the tools differ substantially in meeting the particular organization's needs. Experience indicates that characteristics which are difficult for an organization to evaluate (e.g., lack of access to data, insufficient resources) tend to result in a very narrow range of ratings for the candidate tools. For example, if an organization is not in a position to critically evaluate some characteristic, e.g., reliability, the candidate tools will most likely be assigned ratings in a narrow range (e.g., 3.0 - 3.5 out of 4). The higher the weight assigned to such a characteristic, the smaller will be the spread between the weighted scores, and thus the less the process will be able to discriminate between tools.

Thus the user of this International Standard should assign the highest weights to characteristics which reflect the organization's actual needs and which can be evaluated with some degree of detail.

B.2 Types of algorithms

There are several general approaches to selection algorithms:

An organization may use an algorithm which leads to a single rating for each candidate tool and then compare the ratings.

An organization may establish upper and lower thresholds within which a tool's rating must fall.

An organization may make a selection based upon a profile of each tool and the use of management judgement.

Examples of several typical approaches are provided below.

Algorithms commonly in use include cost-based algorithms, score-based algorithms, and rank-based algorithms. They are discussed below, and examples are provided. There are advantages and disadvantages to each approach - no recommendation is made. Organizations desiring an honest evaluation and selection process should choose an algorithm which it has sufficient resources to implement, and which is appropriate to the specific case in point.

B.3Cost-based algorithms

These algorithms identify some minimal acceptable level of capability (based upon the needs) and identify all tools providing that capability. The acceptable tools are then ranked according to cost. The lowest cost, acceptable tool is presumably recommended. This approach is sometimes mandated by organizational procurement regulations.

Proponents of a cost-based approach usually cite the ease of application, perceived fairness (more objective), and lowest resulting cost of recommended tool.

Opponents of this approach often counter that the precise definition of the actual user requirements is very difficult, and represents a major risk. Another criticism is that this approach is not sensitive to cost/benefit tradeoffs. That is, the low cost tool will meet the minimum requirements, but a tool which is somewhat more expensive might, by exceeding the minimum requirements, provide a substantially higher level of productivity, resulting in a better overall value in the long term.

B.4 Example of the application of a cost-based algorithm

An organization decides that it wants to provide its software developers with a detailed design tool which will allow its users to enter design data and which then produces a data dictionary, certain specific charts and diagrams, and performs a number of consistency and completeness checks (all well-defined). The tool to be purchased must operate in a specific hardware/software environment. The organization identifies all the candidate tools which run in the desired environment and claim to provide the required capabilities. It obtains evaluation copies of the candidates, assigns evaluation personnel to verify that the tools do indeed provide the required products in a satisfactory manner. The costs of each candidate are then computed. This organization considers "cost" to include purchase price, five years of maintenance and updates, documentation, and initial installation and training. The tool whose overall cost is lowest is purchased.

B.5 Score and rank-based algorithms

Score and rank-based algorithms are very similar in that in each case, a single value for each tool is calculated by multiplying the weight given to each user need by some number and adding those products. In the case of a score-based algorithm, the number which is weighted is a score of how well the tool meets the user need, according to some predefined scale (e.g., 4 on a scale of 1 to 5). In the case of a rank-based algorithm, the number which is weighted is the ordinal rank of how well this tool meets the need when compared to the other tools under consideration (e.g., second best out of five candidates). Score-based algorithms attempt to provide an absolute measure of each tool evaluated, and tools can be individually evaluated. The tool with the highest score is recommended. Rank-based algorithms attempt to provide relative measures of a number of tools; tools cannot be individually evaluated, and the result for a given tool is dependent upon the set of competing tools being evaluated. The tool with the lowest score is recommended.

When comparing score and rank-based algorithms to the cost-based approach, proponents of score and rank-based algorithms contend that these algorithms are more sensitive to the needs of tool users: evaluations are typically less functional and more qualitative in nature, and are often performed by future users. They also contend that these approaches are more sensitive to ranges of capabilities (vs. minimum capabilities) and therefore amenable to cost/benefit trade-offs and productivity enhancement analyses.

Proponents of cost-based approaches often argue against score and rank-based algorithms, arguing that the evaluations become much more expensive and subjective. They contend that evaluators allow their personnel biases to direct the assignment of scores or ranks, alleging that evaluators decide first which tool to select (based upon subjective characteristics) and then assign scores or ranks which justify their earlier decision. They also contend that much of the additional detail usually provided in score and rank-based evaluations is specious; that in practice, when the weights are applied and the values from many user needs are combined, the results are usually very close, and the differences in scores which are obtained are somewhat arbitrary.

When comparing score-based algorithms to rank-based algorithms, proponents of score based algorithms point out that score based algorithms provide an "absolute" measure of a tool's quality, and are independent of other tools, in particular, of the specific set of other candidates. They identify as an advantage that score-based evaluations can be carried out independently of one another, using the most appropriate personnel, resources and schedule. Proponents of rank-based approaches counter that evaluating tools in a vacuum is pointless, and that when an organization is evaluating tools with an imminent purchase in mind, direct comparison between the tools is required, and that independence is no advantage. They also point out that while rank-based evaluations require the same personnel to evaluate all the candidate tools, it is very difficult for different evaluators to consistently apply the same evaluation characteristics, making the score for a tool a function of its evaluators, rather than the tool's intrinsic quality.

B.6 Profile algorithms

Consumer product testing agencies often provide results in the form of a profile of each candidate product. In the context of CASE tools, the characteristics of major importance to the user are evaluated and the results input into the selection process. No specific amalgamation of these partial results is done. The selector(s) reviews the profile, and based upon the judgement of the relative importance of the various characteristics to the organization, makes a selection.

B.7 Other applicable algorithms

There are a number of additional selection algorithms, developed in the academic arena, which might be applied when appropriate for the organization. These are particularly useful when an organization is faced with evaluation data which are fuzzy or sparse and is having difficulties in amalgamating the viewpoints of multiple evaluators and/or selectors.

Borda's algorithm (Black, 1958; Fishburn, 1973) - a sum of the ranks algorithm.

Condorcet's algorithm (Black, 1958; Fishburn, 1973) - a pairwise comparison algorithm.

Dodgson's algorithm (Black, 1958; Fishburn, 1973) - a preference measurement algorithm.

Fishburn's algorithm (Fishburn, 1973) - a preference ordering algorithm.

Lexicographic algorithm - a criteria comparison algorithm.

Analytic Hierarchy Process (Saaty, 1980) - a structured algorithm.

The algorithms discussed above are all quantitatively based. Another approach is the qualitatively based approach referred to as Grounded Theory. Rather than starting by identifying requirements to be met and criteria to be measured, this approach begins with an examination of experience to date; e.g., discuss with CASE tool users their experience with CASE technology as a starting point. Relevant references include:

Glasser & Strauss, "*The Discovery of Grounded Theory, Strategies for qualitative research*", Aldine, New York, 1967

Bubcoko, Janis A., Jr., "*Towards a Corporate Knowledge Repository*", SYSLAB Report No. 91-023, 1991

Annex C
(informative)
Bibliography

ISO/IEC JTC1 SC7 Working Group 6 is preparing several documents which are relevant to software quality and evaluation. They include:

Series 1: Software quality characteristics and metrics

ISO/IEC 9126-1:-[1] *Information technology - Software quality characteristics and metrics - Part 1: Quality characteristics and subcharacteristics (Revision of ISO/IEC 9126:1991).*

Series 2: Evaluation of software products

ISO/IEC 14598-1:-[1] *Information technology - Software product evaluation - Part 1: General overview.*

ISO/IEC 14598-2:-[1] *Information technology - Software product evaluation - Part 2: Planning and management.*

ISO/IEC 14598-3:-[1] *Information technology - Software product evaluation - Part 3: Process for developers.*

ISO/IEC 14598-4:-[1] *Information technology - Software product evaluation - Part 4: Process for acquirers.*

ISO/IEC 14598-5:-[1] *Information technology - Software product evaluation - Part 5: Process for evaluators.*

ISO/IEC 14598-6:-[1] *Information technology - Software product evaluation - Part 6: Evaluation modules.*

1) To be published.