

IEEE Recommended Practice for the Adoption of Computer-Aided Software Engineering (CASE) Tools

Sponsor

**Software Engineering Standards Committee
of the
IEEE Computer Society**

Approved December 12, 1996

IEEE Standards Board

Abstract: Difficulties that may be encountered, and how they can be avoided, by organizations intending to adopt CASE tools are addressed. An overview of the adoption process, including analysis of the organization's needs and readiness for automation, use of a pilot project, and definition of activities necessary to integrate the new technology into the organization's standard software engineering practice, is provided.

Keywords: CASE, CASE adoption, pilot project, technology transfer, tool

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

Copyright © 1996 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1996. Printed in the United States of America.

Print: ISBN 1-55937-591-4, SH94373
PDF: ISBN 0-7381-0423-X, SS94373

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

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE 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 technical 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 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 all patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (508) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

[This introduction is not part of IEEE Std 1348-1995, IEEE Recommended Practice for the Adoption of Computer-Aided Software Engineering (CASE) Tools.]

This recommended practice provides recommendations to use in the adoption of computer-aided software engineering (CASE) tools. The term adoption is used in a broad sense. It includes all activities from initial needs evaluation through the full-scale use of the tool(s) in the appropriate parts of the organization. The need for this recommended practice was recognized during the development of IEEE Std 1209-1992, IEEE Recommended Practice for the Evaluation and Selection of CASE Tools. IEEE Std 1348-1995 references IEEE Std 1209-1992 for information regarding CASE tool evaluation and selection.

IEEE Std 1348-1995 does not intend to minimize the challenge of introducing and adopting CASE tools within an organization. In fact, the literature contains many references to attempts at CASE tool adoption that were not successful. In that respect, the following should be noted:

- CASE tools may not be beneficial to every organization that produces software.
- The actual cost of tool adoption is usually much greater than the cost of procuring the tool.
- CASE tools do not necessarily save either time or money in their initial use; savings may only come to be realized over time.
- CASE adoption can be viewed within the broader context of software engineering technology improvement.
- CASE tools provide the potential for substantial benefits when an adoption process is successfully completed.

The following is a list of participants in the CASE Adoption Working Group.

Thomas Vollman, *Chair*

Edwin Morris, *Co-Editor*

Dennis Smith, *Co-Editor*

Syed Ali
Michael Caldwell
Theo Clarke
Barbara Cuthill
Greg Daich

Chris Daly
Sanjay Dewal
Juan Garbajosa-Sopena
Hiranmay Ghosh
Peter Goldsbury

Leonard Green
Fabrizio Imelio
Michael Lackner
Jose Pi
Carl Singer

The following persons were on the balloting committee:

Syed Ali	Robert T. Harley	John G. Phippen
Man K. Au	William Hefley	Dennis Potwora
Oddur Benediktsson	Rick Hefner	Lawrence S. Przybylski
H. Ronald Berlack	Manfred Hein	Kenneth R. Ptack
Christus Bezirtzoglou	Mark Heinrich	John Reddan
Richard E. Biehl	Gary E. Hendrzak	Larry K. Reed
Michael A. Blackledge	Mark Henley	Ann Reedy
William J. Boll	Debra Herrmann	Annette D. Reilly
Kathleen L. Briggs	Umesh P. Hiriyannaiah	Dennis Rilling
Christine Brown	John W. Horch	James R. Roberts
Fletcher Buckley	Jerry Huller	Patricia Rodriguez
Edward R. Byrne	Fabrizio Imelio	Andrew P. Sage
Michael Caldwell	David Johnson	Helmut Sandmayr
Jaya R. Carl	Martha Johnson	Stephen R. Schach
John P. Chihorek	William S. Junk	Gregory D. Schumacher
Theo Clarke	George X. Kambic	Keith Shewbridge
Francois Coallier	Chris F. Kemerer	Robert W. Shillato
Virgil Lee Cooper	Judy Kerner	Carl A. Singer
Gregory T. Daich	Robert J. Kierzyk	James M. Sivak
Taz Daughtrey	Motti Y. Klein	Richard S. Sky
Hillary Davidson	Dwayne L. Knirk	David J. Smith
Neil Davis	Shai Koenig	Michael W. Smith
Raymond Day	Thomas M. Kurihara	Alfred R. Sorkowitz
Sanjay Dewal	Boniface Lau	Richard Staunton
Carl Einar Dragstedt	J. Dennis Lawrence	Fred J. Strauss
Leo G. Egan	Randal Leavitt	Toru Takeshita
Peter L. Eirich	Fang Ching Lim	Tanehiro Tatsuta
Richard L. Evans	William M. Lively	Booker Thomas
Jonathan H. Fairclough	James J. Longbucco	Patricia Trelle
John W. Fendrich	Austin J. Maher	Leonard L. Tripp
Jon J. Fineman	Harold Mains	Mark-Rene Uchida
Jay Forster	Loredana Mancini	Tom Vaiskunas
Richard C. Fries	Gianluca Marcellino	Glenn D. Venables
Juan Garbajosa-Sopena	Robert A. Martin	Udo Voges
Yair Gershkovich	Mike McAndrew	Thomas E. Vollman
Adel N. Ghannam	Patrick McCray	Ronald L. Wade
Hiranmay Ghosh	Sue McGrath	Dolores Wallace
Marilyn Ginsberg-Finner	Jacques Meekel	John W. Walz
John Garth Glynn	Jerome W. Mersky	David Waxman
Julio Gonzalez Sanz	Alan Miller	Camille S. White-Partain
Lewis Gray	Millard Allen Mobley	Scott A. Whitmire
Lawrence M. Gunther	Charles S. Mooney	Charles D. Wilson
David A. Gustafson	James W. Moore	Paul A. T. Wolfgang
Robert P. Hanrahan	R. Muralidharan	Natalie C. Yopconka
John Harauz	Dennis E. Nickle	Weider D. Yu
Rob Harker	Mike Ottewill	Janusz Zalewski
	Donald J. Pfeiffer	

When the IEEE Standards Board approved this standard on December 12, 1995, it had the following membership:

E. G. “Al” Kiener, *Chair*

Donald C. Loughry, *Vice Chair*

Andrew G. Salem, *Secretary*

Gilles A. Baril
Clyde R. Camp
Joseph A. Cannatelli
Stephen L. Diamond
Harold E. Epstein
Donald C. Fleckenstein
Jay Forster*
Donald N. Heirman
Richard J. Holleman

Jim Isaak
Ben C. Johnson
Sonny Kasturi
Lorraine C. Kevra
Ivor N. Knight
Joseph L. Koepfinger*
D. N. “Jim” Logothetis
L. Bruce McClung

Marco W. Migliaro
Mary Lou Padgett
John W. Pope
Arthur K. Reilly
Gary S. Robinson
Ingo Rüsçh
Chee Kiow Tan
Leonard L. Tripp
Howard L. Wolfman

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
Steve Sharkey
Robert E. Hebner
Chester C. Taylor

Kim Breitfelder
IEEE Standards Project Editor

Contents

CLAUSE	PAGE
1. Overview.....	1
1.1 Scope.....	1
1.2 Purpose.....	2
1.3 CASE features and expectations.....	2
1.4 Organizational readiness.....	3
1.5 CASE caveats	4
2. References.....	4
3. Definitions	5
4. Overview of CASE adoption	5
4.1 Definition of successful CASE adoption	5
4.2 Steps in CASE adoption	6
5. Defining CASE needs.....	7
5.1 Analyzing the organization’s capabilities and readiness	7
5.2 Defining organizational needs	8
5.3 Reviewing the CASE tool marketplace	8
5.4 Defining success criteria.....	8
5.5 Developing a CASE adoption strategy	9
6. Evaluating and selecting CASE tools	9
6.1 Evaluation	10
6.2 Selection.....	11
7. Conducting the pilot.....	11
7.1 Determining pilot characteristics	12
7.2 Planning the pilot	13
7.3 Performing the pilot	15
7.4 Evaluating the pilot.....	16
7.5 Adoption decision	17
8. Fostering the routine use of the tool	17
8.1 Developing a migration plan.....	17
8.2 Implementing the migration plan.....	19
ANNEX	
Annex A (informative) Organizational analysis	21
Annex B (informative) Defining organizational CASE goals, needs, and expectations	24

Annex C (informative) Criteria for success	28
Annex D (informative) Approaches to developing a CASE tool adoption strategy	29
Annex E (informative) Evaluating the pilot.....	31
Annex F (informative) Fostering routine use of the tool	33
Annex G (informative) Bibliography	35

IEEE Recommended Practice for the Adoption of Computer-Aided Software Engineering (CASE) Tools

1. Overview

This recommended practice was created to aid organizations in the successful adoption of computer-aided software engineering (CASE) tools. Many software development organizations have turned to CASE technology in an attempt to improve the manner in which software is developed. However, in spite of the perceived potential of CASE technology, some organizations do not achieve the expected benefits.

While there are many potential reasons for the failure of CASE technology to produce the desired results, a significant part of the discordance between expectations and reality seems to be founded upon an inadequate understanding of the development process and CASE adoption practices. These are critical for the long term success of CASE technology within the organization.

1.1 Scope

This recommended practice views the CASE adoption process as more than the selection of CASE tools. The CASE adoption process consists of the activities by which an organization identifies appropriate CASE tools and related technologies and introduces them into general use. The steps of a successful CASE adoption process include the following:

- a) Defining CASE needs
- b) Evaluating and selecting CASE tools
- c) Conducting a pilot
- d) Fostering routine use of the tool

A successful CASE adoption process is not limited solely to tool use. Rather, it embraces the planning and implementation of an entire set of technical, organizational, cultural, and management processes necessary to achieve the required organizational improvements. It is founded upon a strong understanding of what CASE tools can and cannot do for the organization.

Steps a) through d) above are appropriate for a wide range of computing organizations, including those that develop management information systems and those that develop scientific and technical systems. This recommended practice can also be of benefit to government and commercial software organizations.

Specific situations may influence the manner in which CASE adoption is performed. For example, when a specific tool is mandated by customer preference or contract requirements, adoption steps will require tailoring to reflect the predetermined nature of tool selection. In other cases, the relative simplicity or

complexity of the tool, the degree to which the tool supports or conflicts with existing processes, the required degree of integration with other tools, and the characteristics and experience of the targeted users, may lead to the tailoring of the adoption process.

The impact of tool adoption on an organization will also vary according to the specific situation. For example, many organizations adopt CASE tools in order to automate current processes and thereby increase productivity and quality. In this situation, changes in the organization's processes may be limited to those necessary to make them amenable to automation, or they may be broader in scope.

Other organizations may select tools that integrate well with existing tools. This approach can simplify the adoption process. However, attempts to adopt an entire suite of tools can be problematic, since the complexity of the adoption process is greatly increased by wide scale changes to tool support.

Still other organizations may choose to adopt CASE tools in order to enable the introduction and use of other technologies, such as formal or structured design methods. In this situation, CASE adoption occurs in parallel with significant process modification.

Finally, while this recommended practice is written from the perspective of adopting a new category or type of tool, it can also be tailored to assist organizations replacing existing CASE tools with alternate tools.

1.2 Purpose

The purpose of this recommended practice is to provide guidance that will enhance the probability of the successful adoption of CASE technology. The use of this recommended practice will help maximize the return and minimize the risk of investing in CASE technology.

Regardless of the reason for CASE adoption, it is important to recognize that CASE adoption is not amenable to a cookbook approach, which guarantees dramatic improvement in an organization's software development and maintenance activities. Rather, the adoption process implemented should reflect the particular needs of the organization. In return, incremental improvements in the organization's software development and/or maintenance activities can be expected.

1.3 CASE features and expectations

The range of capabilities considered to be CASE has grown significantly since the early application of the term to tools providing support for structured analysis and design. Some CASE tools provide support for specific phases of the life cycle. Examples include analysis and design tools, code generators, and testing tools. Other CASE tools provide functionality across the life cycle. Such tools include project management tools, configuration management tools, and documentation tools. Currently the vision for a full software engineering environment is that of an integrated set of CASE tools that supports all aspects of the software development process.¹

Regardless of the category and features of the tool, some CASE users claim significant gains from using the CASE tools. These gains include the following:

- Increased developer productivity
- Improvements in the quality of the delivered software
- Improved consistency and uniformity of the development approach

There is also a general belief that CASE tools can mitigate such problems as the following:

- Inadequate specification of requirements
- Unreliable delivered software

¹A reference model describing the services that might be found in a full software engineering environment is available in [B20]. An overview of framework services to support such an environment is available in [B19]. See annex G.

- Deliveries that are late and/or over budget
- Insufficient capability to detect inadequate designs
- Insufficient and inaccurate documentation leading to poor maintainability
- Poor development methods
- Lengthy test cycles and inadequate testing
- Inconsistencies in the software development process

Because of the diverse nature of CASE tools, it is misleading to make blanket statements about what CASE actually offers relative to expectations. Some tools, such as configuration management tools and documentation tools, are generally accepted mechanisms for improving the manner in which software is developed. Not as well understood and accepted is the impact of other tools that provide analysis and design, reverse engineering, code generation support, and testing support.

Factors that complicate the determination of the impact of CASE include the following:

- The wide variation in quality and capability within a single category of tools
- The relatively short time that many types of CASE tools have been in use in organizations
- The wide difference in the adoption practices of various organizations
- The general lack of detailed metric data for previous and current projects
- The wide range of project domains
- The compounding impact of changes to methods and processes that are often associated with the adoption of CASE tools
- Personnel's lack of experience with appropriate methods
- Variations in the degree of tool integration within different organizations

As expected, based on the difficulty of determining the impact of CASE tools, the data available are extremely limited and vary widely. The data (primarily from case studies of organizations adopting CASE tools) suggest that organizations have experienced variations in the impact of CASE tools depending on the type of tool, project characteristics, the degree of management support, and the sophistication of the users.² In addition, some CASE analysts expect that true gain for some types of tools is only realized after one to two years of experience. Others suggest that the impact may only become evident during the maintenance phase of the software life cycle, where improved software design reportedly attributable to CASE technology may lead to lower maintenance costs. In addition, the lack of consistency in gathering this limited data further frustrates comparison.

1.4 Organizational readiness

The key to the successful adoption of a CASE tool is the readiness of the organization. Readiness of the organization involves:

- *Technology.* Recognition of current capabilities and the ability to incorporate new technology
- *Culture.* Willingness to adapt to new processes and interactions
- *Management.* Leadership and commitment to a substantial process

Regardless of how carefully the various recommendations for CASE adoption are applied, without sufficient readiness in these areas, the CASE adoption effort is likely to end in frustration. A number of pitfalls resulting from insufficient readiness are discussed in 1.5.

²Additional information concerning CASE tool benefits and pitfalls can be found in [B1], [B6], [B7], [B17], and [B25].

1.5 CASE caveats

In order to make an informed decision concerning whether to invest in CASE technology, an organization is forced to make assessments about the value of individual tools based on inconsistent and inconclusive data. This problem is often compounded by inexact knowledge of potential pitfalls of CASE use within the organization. Among the more vexing issues are the following:

- *Assessment of return on investment in CASE tools.* The actual value of CASE tools is difficult to determine due to the lack of available metrics data on software development projects and processes.
- *Management commitment to tool use.* The adoption of CASE tools can be a long term process, and may not produce immediate results. Management, therefore, may not remain committed to CASE tools.
- *Investment of resources.* The cost of adopting CASE technology is higher than many organizations anticipate.
- *Current processes and methods.* The frequently inexact match between the processes and methods supported by CASE tools and those utilized within the organization can be troublesome.
- *Tool scalability.* Many CASE tools have limited capabilities in dealing with very large systems.
- *Tool integration.* CASE tools are frequently difficult to use in conjunction with one another. Difficulties arise both from the different paradigms supported by the tools and in the transfer of data and control from one tool to another.
- *Tool overhead.* Some CASE tools require too much effort to justify their use in a small development activity that could otherwise benefit from the discipline that CASE can impose.
- *Personnel.* The sometimes negative attitude of personnel towards adopting new CASE tool technology can be a major contributor to the failure of a CASE adoption effort.
- *Support mechanisms.* The expense of CASE tools and of keeping personnel skills current is often more than anticipated.

The decision to bring a CASE tool into an organization should be made based on the expected return on investment. This requires an awareness of the potential benefits (e.g., improved productivity and quality) and risks of tool adoption. Many organizations underestimate the need for training, user support, and definition of new procedures that are required for successful use of CASE tools. Among the problems are the following:

- Short term decrease in productivity as a result of the adoption effort
- Dissatisfaction on the part of employees adopting the new technology
- Difficult changes to process and methods
- Complex integrations with existing tools
- Extensive training requirements
- Significant expense

CASE users must address long-term maintenance costs, frequent releases and the potentially rapid obsolescence of tools, and the ongoing costs of training new staff and upgrading the skills of existing staff.

2. References

This standard shall be used in conjunction with the following standards. When the following standards are superseded by an approved revision, the revision shall apply.

IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology (ANSI).³

IEEE Std 1002-1987 (Reaff 1992), IEEE Standard Taxonomy for Software Engineering Standards (ANSI).

³IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.

IEEE Std 1209-1992, IEEE Recommended Practice for the Evaluation and Selection of CASE Tools (ANSI).

ISO/IEC 9126: 1991, Information technology—Software product evaluation—Quality characteristics and guidelines for their use.⁴

3. Definitions

This clause contains terms as they are used in this recommended practice.

3.1 CASE tool: A software tool that aids in software engineering activities, including, but not limited to, requirements analysis and tracing, software design, code production, testing, document generation, quality assurance, configuration management, and project management.

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

3.2 computer-aided software engineering (CASE): The use of computers to aid in the software engineering process. May include the application of software tools to software design, requirements, tracing, code production, testing, document generation, and other software engineering activities.

3.3 software engineering environment: The hardware, software, and firmware used to perform a software engineering effort. Typical elements include computer equipment, compilers, assemblers, operating systems, debuggers, simulators, emulators, test tools, documentation tools, and database management systems.

NOTE—For this recommended practice, a software engineering environment is a collection of CASE tools. These tools are brought together in support of the methods, policies, guidelines, and standards that comprise the activities of a software engineering process. Software engineering environments display varying levels of integration. Integrating mechanisms may take many forms, such as a database, a message broadcast system, an agreed transfer protocol, or a common schema.

4. Overview of CASE adoption

4.1 Definition of successful CASE adoption

A successful CASE adoption effort will:

- a) Provide an appropriate level of technological support to the software development and maintenance processes
- b) Have a positive impact on some or all of the following:
 - 1) Productivity
 - 2) Product quality and defect rates
 - 3) Adherence to standards
 - 4) Documentation
 - 5) Ease of quantitative measurement
- c) Attain an appropriate level of return on investment for the organization
- d) Lead to the general and customary use of the tool or tools within the organizations or groups for which they were intended

⁴ISO/IEC publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembe, CH-1211, Genève 20, Switzerland/Suisse. They 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.

4.2 Steps in CASE adoption

The CASE adoption process identifies four steps and an interrelated set of activities for each of the steps. Steps are illustrated in figure 1 and discussed in subsequent subclauses. Additional information to aid in the planning and execution of these steps and activities is found in the annexes.

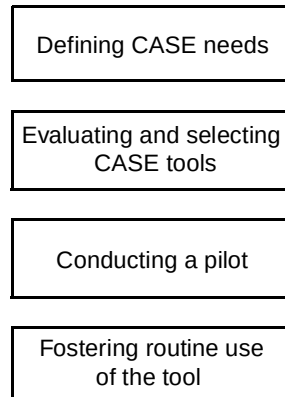


Figure 1—CASE adoption process

4.2.1 Step 1: Defining CASE needs

This step involves acquiring an understanding of the needs of the organization, the technology available, and the CASE adoption process to be followed. Defining CASE needs will lead to an identification of areas where CASE can improve an organization's practice. The outcome of this step is a document defining the CASE adoption strategy. Specific activities are discussed in clause 5 and illustrated in figure 2.

4.2.2 Step 2: Evaluating and selecting CASE tools

IEEE Std 1209-1992⁵ provides a recommended practice for evaluating and selecting CASE tools. Activities for tool evaluation include the following:

- Preparing a statement of the purpose of evaluation
- Identifying the evaluation criteria
- Identifying candidate tools
- Evaluating the candidate tools
- Reporting the results of the evaluation process

The selection process interacts closely with the evaluation process. Activities include the following:

- Preparing a selection task definition statement
- Performing selection activities
- Reporting the results of the selection process

The outcome of step 2 is a recommendation for the selection of one or more CASE tools.

4.2.3 Step 3: Conducting a pilot

Prior to full introduction of the recommended tool into the organization, a pilot is conducted in order to verify the decisions made in steps 1 and 2, and to prepare for the introduction. The activities involved in the pilot are discussed in clause 7 and shown in figure 4.

The outcome of step 3 is a document that discusses the results of the pilot and details the adoption decision.

⁵Information on references can be found in clause 2.