Current utilization of CASE technology: lessons from the field

Industrial Management + Data Systems; Wembley; 2000; Mark E. McMurtrey;James T.C. Teng;Varun Grover;Hemant V. Kher;

Volume: 100
Issue: 1
Start Page: 22-30
ISSN: 02635577
Subject Terms: Computer aided software engineering
Systems development
Automation
Studies
Classification Codes: **5240**: *Software & systems*
7500: *Product planning & development*
9130: *Experimental/theoretical*

Abstract:
Since its inception, CASE (computer-aided software engineering) tools have been hailed as the silver bullet of applications development. Although these tools have failed to live up to such an advance billing (as do most fix-all solutions), these products remain a viable option for practitioners of modern applications development. This study comments on the use of CASE in modern IS installations, using the results of an in-depth survey completed by 226 IS professionals from over 30 Fortune 500-type companies. First, the study identifies the most popular features possessed by respondents' CASE toolsets. Next, it comments on the gap perceived to exist between CASE features actually possessed, and those features needed by these professionals in the performance of their job duties. Finally, implications for practice and research are presented.

Full Text:

Mark E. McMurtrey: Francis Marion University, Florence, South Carolina, USA

James T.C. Teng: University of South Carolina, Columbia, South Carolina, USA

Varun Grover: University of South Carolina, Columbia, South Carolina, USA

Hemant V. Kher: University of South Carolina Sumter, Sumter, South Carolina, USA

Introduction

While certainly not the "silver bullet" to eradicate the information system (IS) organization's applications backlog, CASE (Computer-aided software engineering) technology remains a viable option for those responsible for the development and timely delivery of modern-day ISs. Like many other alternatives to the snail's pace of progress during the traditional systems development life-cycle (SDLC), such as prototyping, rapid application development, the use of canned software packages, or reusable code, CASE tools have great potential in alleviating (not necessarily eliminating) the applications backlog. Although the latest, greatest (or most publicized) advancement seems to be the proliferation of object-oriented analysis and design (OOAD) techniques (Freeman, 1997; Post et al., 1997), it is still unclear as to whether these OOAD tools will become the final answer to the ever-increasing problem of delivering application systems in a more timely manner (c.f. Johnson and Hardgrave, 1997; Wu and Wu, 1994).

Resolving the issue of what technique will ultimately reign superior in diminishing application portfolio backlogs is beyond the scope of this paper, as such a debate is contingent on the peculiarities and environmental culture associated with each unique IS development unit. Rather, this paper outlines lessons learned in the current utilization of conventional CASE technology, from an extensive investigation into the use of these tools by 226 IS personnel representing a wide range of industries and experience.

The purposes of this research are twofold. First, an extensive exploration into the most popular CASE tool features possessed by respondents is reported. Second, the paper will concentrate on the perceived "gap" between specific CASE tool features possessed by participants in this study, and those features perceived to be needed by said participants in the performance of their job duties. The results of this investigation will provide a prescription for IS managers interested in acquiring CASE technology for their IS department, as well as setting up a framework for more intensive studies into CASE use by the research community. Before proceeding, however, some background information about CASE is necessary.

Background

Simply put, CASE is the automation of "anything a human does to software" (Stamps, 1987). The broadest definition of a CASE tool is any software tool that provides automated assistance for software development, maintenance, or project management activities (McLure, 1989). From its inception in the early to mid-1980s, CASE has evolved from simple "upper-CASE" tools such as screen and report painters, and "lower-CASE" tools such as code generators, to completely integrated CASE facilities that support the entire range of the SDLC. Integrated CASE (or ICASE) is a set of integrated CASE tools sharing a common user interface, data interface, life-cycle framework, and repository (du Plessis, 1993). The following section outlines some common advantages and disadvantages associated with using CASE tools.

CASE technology: benefits and drawbacks

Besides its overall potential to alleviate the applications systems development bottleneck, CASE tools possess many features and functionalities that contribute to improving the performance of systems analysts and designers. Chief among these advantages include diagramming aids for drawing architectural representations of systems and system components (e.g. data flow diagrams, entity-relationship diagrams, structure charts); an integrated encyclopedic repository for all data element definitions and pictorial representations; and code generators to generate executable code from proposed system specifications.

Disadvantages of employing CASE tools in the systems development process include high start-up costs for hardware and training; a reliance on structured methodologies (SDLC); poor integration among different vendor products; and a long learning curve. The list below shows some of the more common benefits and drawbacks associated with CASE, some of which were adapted from doctoral theses involving CASE by Bellin (1991), McMurtrey (1997), Nelson (1991), Norman (1987), Ongkasuwan (1991), and Rai (1991).

Benefits

CASE technology:

- increases productivity and quality;

- enables uniformity among the design team during the analysis and design process;

- enables ease of making changes during the various SDLC phases;

- can restructure poorly written code;

- simplifies programming maintenance;

- enforces software/systems engineering standard;

- aids project management;

- provides for a central data repository and reference encyclopedia;

- simplifies drawing a system' s artchitectural diagrams;

- decreases application development time;

- provides the capability to solve larger and more complex problems; and

- generates code automatically.

Drawbacks

CASE technology:

- may cause extreme amount of sales and press hype;

- leads to reliance on structured methodologies;

- necessitates a long learning curve;

- has limited functions;

- does not provide integrated central repository;

- may cause poor integration among different vendor products;

- requires working knowledge of underlying methodology;

- may cause user resistance;

- may cause a change in job roles and relationships and the quality of work life; and

- has benefits which are difficult to measure.

Although increasing productivity is hailed by vendors as the ultimate goal of using CASE, research into this issue has been met with mixed results. Evidence of this discrepancy can be seen by comparing studies that found CASE usage increasing productivity (Subramanian and Zarnich, 1996; Freeman, 1992; Ho, 1992; Banker and Kauffman, 1991; Zagorsky, 1990; Norman and Nunamaker, 1989; Necco et al., 1989), decreasing productivity (Selamat et al., 1994; Ongkasuwan, 1991; Loh and Nelson, 1989), and having no effect on productivity (Vessey et al., 1992; Hayley and Lyman, 1990; Card et al., 1987). Until more refined measurement techniques are available to investigate the productivity debate, this issue will be left as an avenue of further research.

Functional CASE technology model

One of the primary motivations of this study was to empirically investigate the use of CASE technology within the context of Henderson and Cooprider's (1990) functional CASE technology model (FCTM). This theoretically-based model was established from a systematic evaluation of CASE features by expert practitioners. The FCTM is composed of the five dimensions of representation, analysis, transformation, control, and cooperative functionality (for further discussion of the FCTM, the reader is directed to the original manuscript by Henderson and Cooprider, 1990), each containing an appropriate number of features to examine each component. Subjects were requested to respond to a series of three questions each that pertained to the possession, need, and use of a particular feature. Although not all features or components of the model will necessarily be known to all respondents, the FCTM should be representative of modern CASE installations.

Methodology

The current study is part of an ongoing investigation into the use of CASE in IS development units. The researchers developed an in-depth survey instrument designed to capture the perceptions of those closest to the technology: the users, or applications developers, themselves. The questionnaire was designed using demonstrated research constructs with desirable psychometric properties. Specific questions pertaining to CASE features were adapted from the FCTM, developed by Henderson and Cooprider (1990) from an MIT study. The FCTM was refined through a Q-sort analysis, and extensive consultation, with vendors and practitioners of CASE technology.

The 226 respondents in this study come from a wide variety of backgrounds and experience. Table I shows that almost 30 percent were from the insurance industry, while just under one-quarter (23.5 percent) were from manufacturing concerns. Table II indicates that almost one-half (47.8 percent) of the respondents had between 8 and 17 years of experience in the IS field, while the average age was 38 years (table not shown).

Thus, from the perspective of age and experience, the 226 respondents appear to be mature members of their profession. Therefore, the sample frame is (to the extent we can determine) representative of our target population: users of CASE tools. It is hoped that experienced IS developers, such as those that participated in this research, would respond conscientiously and responsibly to questions in the survey instrument.

In the Appendix, Tables AI and AII are used initially in the discussion of the FCTM. A key is provided following these tables that lists each of the 58 items, grouped by FCTM dimension. Afterwards, Table AIII lists the top ten features, grouped by FCTM dimension, of CASE technology possessed by the respondents.

Discussion

FCTM data analysis

Tables AI and AII present the raw data from which discussion of the FCTM is based. Table AI depicts the frequency counts for each of the 58 items in the FCTM section of the research instrument. These tabulations are from all 226 respondent questionnaires. Table AII portrays the percentage measure of these frequency counts (calculated as the frequency count divided by 226). The first column indicates possession, the second column need, and the third column, use. The present discussion focuses on the first two columns.

The top two or three features in each FCTM dimension (e.g. representation, Analysis, etc.) are marked with three asterisks (***). In addition, boxes have been drawn around the raw data figures. Other highlighted observations (to be discussed in the following section) are marked with a single asterisk (*) and also have boxes drawn around figures of interest. Inspection of the two tables indicates that, overall, the feature most often cited as being possessed is described in question number 2, the ability to represent a design in terms of data models. This is followed by question 1, the ability to represent a design in terms of process or flow models. Both of these questions are from the Representation component of the FCTM (questions 1-9). Among those features pertaining to the Analysis component (questions 10-16), the ability to detect inconsistencies in models, definitions, etc. (question 13) was cited most often, followed by a tie between the ability to test for consistency between a process model and a data model (question 10) and identifying where predefined criteria or rules have been violated (question 24). Among the items representing the Transformation component (questions 27-37), the ability to generate screen mock-ups (question 36) was cited most often, followed by providing documentation as a by-product of design (question 34) and generating executable code from a screen mock-up (question 27).

The ability to "freeze" a portion of a design to protect it from changes (question 44) and specifying who can modify various parts of the design work (question 43) were the most often cited features of the Control component of the FCTM (questions 38-49). Allowing concurrent use by several users of dictionary/diagram/etc. (question 53) and allowing a group of users to work simultaneously on a single task (question 51) were the features cited most often from the Cooperative Functionality component of the FCTM (questions 50-58). Table AIII lists the top ten features, grouped by FCTM dimension, of CASE technology possessed by the respondents. The observed frequency (from Table AI) and percentage (from Table AII) for each question is shown, along with a description of the scale item.

FCTM data analysis - the "gap"

Whereas the preceding narrative was in terms of functions possessed, additional data pertaining to the extent (small or large) of need, as well as use, of these features were gathered. The functions most often identified as possessed were also, for the most part, the same functions identified most as being needed, and used, by the respondents. There were, however, a few notable observations. These anomalies are marked with a single asterisk (*), and have boxes drawn around the figures of interest, in Tables AI and AII.

Question number 20 (suggest problem resolutions based on previously used solutions), an Analysis component, was possessed by only 14 of the respondents. However, 137 persons cited this feature as being needed to "some" or "large" extent. A similar phenomenon occurs with question number 35 (perform reverse engineering), a Transformation component. A total of 62 respondents indicated that their CASE installation possessed this feature. However, 130 persons cited it as being needed to "some" or "large" extent. Question number 49 (remind team members about approaching deadlines), part of the Control dimension, was possessed by a mere 24 respondents. Nonetheless, 121 persons needed it to "some" or "large" extent. Question number 56 (allow giving of anonymous feedback or input), a Cooperative Functionality component, was possessed by only six respondents, although 71 persons indicated, to "some" or "large" extent, a need for it.

On one hand, perhaps these are features that some CASE vendors will need to find worthy of inclusion into future versions of their products. On the other hand (and perhaps a more likely scenario), it behoves the organizations for whom respondents work to investigate the acquisition of more sophisticated (at least in terms of technologically advanced) CASE technology that already contains these needed features.

Implications

As mentioned in the introduction, a major portion of this part of our ongoing research program into CASE use is in the "gap" between CASE features possessed by respondents and those perceived to be needed by these practitioners. For instance, one interesting statistic to report would be the "top ten" CASE tool features possessed by the respondent' s CASE toolset. Table AIII presents such an analysis, indicating the two most frequently cited features per FCTM dimension. However, this table alone does not indicate the discrepancies that exist between needed features and those actually possessed.

For example, there is a gap (as shown in Tables AI and AII) between the 14 respondents that reported possessing the feature described in question 20 (suggest problem solutions based on previously used solutions) and the 137 (102 + 35) respondents that indicated a need, to some extent or large extent, for this feature. The implication of such a finding could be that modern-day developers of ISs using CASE would find the inclusion of such a feature as paramount to improving their efficiency and effectiveness at delivering timely solutions. Vendors of CASE technology should ensure that their product incorporates such a feature, while practicing IS managers ascertain whether this feature is viable for their own IS development environment. Thus, such results impact the continuum ranging from both vendors and developers of CASE technology to, ultimately, end users of CASE technology. Furthermore, future research into CASE use might investigate whether such features have been incorporated into newer releases of the software, and how such features impact productivity and efficiency.

A similar rationale may be employed concerning question 35 (perform reverse engineering), question 49 (remind team members about approaching deadlines), and question 56 (allow giving of anonymous feedback or input). These questions are highlighted because of the gaps existing between those respondents' tools that already possess these features and the perceived need for these features indicated by an even greater number of subjects. The latter two features appear to be oriented toward group or team effectiveness, and as such could fall under the umbrella of "groupware" or other team-based technologies. Either way, from the results reported in this research, there appears to be a great need for some of these features (more so than others) by software developers using CASE technology.

Conclusion

This study has reported on the use of CASE technology by 226 IS professionals from over 30 Fortune 500-type companies. The current inspection focused on the most popular CASE tool features possessed, as well as the perceived "gap" between features possessed, and those needed, by respondents. Considerable insight was obtained on the current utilization of CASE, especially in terms of which features seem to be the most popular. Furthermore, a dynamic relationship was uncovered between an existent "gap" of features needed versus those actually possessed. Future research into CASE should attempt, among other things, to determine the extent to which this gap has been bridged.

References

1. Banker, R.D. and Kauffman, R.J. (1991), "Reuse and productivity in integrated computer-aided

software engineering: an empirical study", Management Information Systems Quarterly, Vol. 15 No. 3, September, pp. 375-401.

2. Bellin, D. (1991), "Information commodities and the production process: how the introduction of CASE tools affects the software development process", unpublished doctoral dissertation, The City University of New York, New York, NY.

3. Card, D.N., McGarry, F.E. and Page, G.T. (1987), "Evaluating software engineering technologies", IEEE Transactions on Software Engineering, Vol. SE-13 No. 7, July, pp. 845-51.

4. du Plessis, A.L. (1993), "A method for CASE tool evaluation", Information and Management, Vol. 25, pp. 93-102.

5. Freeman, E. (1997), "Is object oriented application development worth the cost?", Datamation, Vol. 43 No. 3, pp. 82-6.

6. Freeman, R.J. (1992), "An empirical study of the linkage of CASE, function points, and systems development", in Forte, G., Madhavi, N.A. and Muller, H.A. (Eds), Proceedings of the Fifth International Workshop on Computer-Aided Software Engineering, IEEE Computer Society Press, Los Alamitos, CA, pp. 254-7.

7. Hayley, K.J. and Lyman, H.T. (1990), "The realities of CASE", Journal of Information Systems Management, Vol. 7 No. 3, Summer, pp. 18-23.

8. Henderson, J.C. and Cooprider, J.G. (1990), "Dimensions of I/S planning and design aids: a functional model of CASE technology", Information Systems Research, Vol. 1 No. 3, September, pp. 227-54.

9. Ho, Y. (1992), "To what extent will CASE tools assist users in the systems development? - A case study in academic environment", in Lederer, A.L. (Ed.), Proceedings of the 1992 ACM SIGCPR Conference, ACM Press, Cincinnati, OH, pp. 93-6.

10. Johnson, R.A. and Hardgrave, W.C. (1997), "Evaluation of object-oriented systems design (OOSD): opinions, cases and empirical research", Proceedings of the 28th Annual Meeting of the Decision Sciences Institute, San Diego, CA (in press).

11. Loh, M. and Nelson, R.R. (1989), "Reaping CASE Harvests", Datamation, 1 July, pp. 31-4.

12. McClure, C. (1989), "The CASE experience", Byte, Vol. 14 No. 4, April, pp. 235-44.

13. McMurtrey, M.E. (1997), "Determinants of job satisfaction among systems professionals: an empirical study of the impact of CASE tool usage and career orientations", unpublished doctoral dissertation, University of South Carolina, Columbia, SC.

14. Necco, C.R., Tsai, N.W. and Holgeson, K.W. (1989), "Current usage of CASE software", Journal of Systems Management, Vol. 40 No. 5, May, pp. 6-11.

15. Nelson, A.C. (1991), The Effects of CASE Tools and Systems Development Methodology on Systems Analysts, unpublished doctoral dissertation, University of Pittsburgh. Pittsburgh, PA.

16. Norman, R. J. (1987), "Integrated development environments in support of information systems design methodologies and systems analysts' productivity", unpublished doctoral dissertation, University of Arizona. Tuscon, AZ.

17. Norman, R.J. and Nunamaker, J. (1989), "CASE productivity perceptions of software engineering professionals", Communications of the ACM, Vol. 32 No. 9, September, pp. 1102-8.

18. Ongkasuwan, M. (1991), "An investigation of the effects of computer-aided software engineering tools, system complexity, and systems analyst's experience on the system design quality and productivity: a laboratory experiment", unpublished doctoral dissertation, Georgia State University. Atlanta, GA.

19. Post, G., Kagan, A. and Keim, R.T. (1997), "Object-oriented (OO) CASE tools: an evaluation of ROSE", Proceedings of the 28th Annual Meeting of the Decision Sciences Institute, San Diego, CA, (in press).

20. Rai, A. (1991), "Stimulating the use of computer-aided software engineering in information systems departments: an empirical test of elements of innovation theory", unpublished doctoral dissertation, Kent State University. Kent, OH.

21. Selamat, M.H., Choong, C.Y., Othman, A.T. and Rahim, M.M. (1994), "Non-use phenomenon of CASE tools: Malaysian experience", Information and Software Technology, Vol. 36, No. 9, pp. 531-7.

22. Stamps, D. (1987), "CASE: cranking out productivity", Datamation, 1 July, pp. 55-8.

23. Subramanian, G.H. and Zarnich, G.E. (1996), "An examination of some software development effort and productivity determinants in ICASE tool projects", Journal of Management Information Systems, Spring, Vol. 12 No. 4, pp. 143-60.

24. Vessey, I., Jarvenpaa, S.I. and Tractinsky, N. (1992), "Evolution of vendor products: CASE tools as methodology companions", Communications of the ACM, April, Vol. 35 No. 4, pp. 90-105.

25. Wu, S. and Wu, M.S. (1994), Systems Analysis and Design, West Publishing Company, St. Paul, Minneapolis.

26. Zagorsky, C. (1990), "Case study: managing the change to CASE", Journal of Information Systems Management, Vol. 7, No. 3, Summer, pp. 24-32.

[Illustration]

Caption: Table I; Response by industry type; Table II; Years in information systems field; Table AI; Response frequency counts for all 58 features; Table AIb; Response frequency counts for all 58 features; Table AII; Percentage response for all 58 features; Table AIIb; Percentage response for all 58 features; Table AIII; Top ten features possessed (two per dimension); Table AIV; Key item description for the 58 features of the FCTM