

# A Study on the Evaluation of Case Technology

HASSAN ELSHAZLY

AND

VARUN GROVER

UNIVERSITY OF SOUTH CAROLINA

## ABSTRACT

Little is known about the actual usage of Computer Aided Software Engineering (CASE) in American organizations and more importantly, whether it is successful. This article attempts to clarify the CASE concept and synthesize the proclaimed benefits of CASE into a testable framework. It then reports results from an exploratory study. Data from 53 organizations is gathered in two phases to determine how the actual benefits being realized from the technology compare with a-priori expectations. The results suggest a number of barriers to the adoption, implementation and success of CASE. Prescriptive implications for IS managers wishing to successfully adopt and implement CASE are discussed.

## INTRODUCTION

The use of Computer Aided Software Engineering (CASE) tools has proliferated over the past five years. Although numerous articles advocating the use of CASE have been published, primarily in practitioner based outlets [2,9,10,11,18,35], very little is known about the actual usage of CASE in American organizations, and more importantly, whether a-priori expectations and proclaimed benefits are being realized. This study attempts to provide an overview of its usage and evaluation. A number of questions are examined. Specifically,

- (a) What is CASE?
- (b) What are the proclaimed benefits organizations can realize through CASE?
- (c) What benefits are being realized by companies that are using CASE?
- (d) What perceptions exist regarding benefits of CASE in organization that have never used CASE?
- (e) How does this compare to companies that once used and abandoned CASE?

In sum, the paper attempts to identify requirements for successful CASE implementation by analyzing a-priori expectations and realization of benefits in various organizations.

Data for this study was gathered in a two phase investigation. First, questionnaire data on realization of benefits was gathered from 53 organizations. Second, In-depth interviews were conducted in 5 of the organizations to gain

further insight into CASE implementation.

It was believed that analysis of the survey and the follow up discussions would reveal two sets of barriers that needed to be overcome in order to successfully adopt/permanently introduce CASE into an organization. The first set are the barriers to adoption. These are factors that are perceived by the organization as being the advantages/disadvantages of using CASE. At a certain point in time the organization will perceive that the advantages outweigh the disadvantages and will acquire CASE. As the adoption process begins a second set of barriers will be encountered, which are the barriers to implementation. These barriers are considerable [13]. They are mainly a result of internal factors within the organization, and in the final analysis determine whether CASE succeeds or fails within the organization.

The results of this study will not only provide a better understanding of the CASE adoption/implementation process, but can also provide prescriptions to IS managers wishing to adopt/implement CASE. Since CASE is not a monolithic concept [8] and can be implemented at many levels, it becomes necessary to formally define the concept in order to place this research in focus.

## CASE DEFINED

CASE stands for Computer Aided Software Engineering. Software Engineering is defined as being "The establishment and use of sound engineering principles (Methods) in order to obtain economical software that is reliable and

works on real machines" [2]. Software Engineering is more than simply converting a set of program specifications to executable code. It is a process that spans the entire software life cycle from the moment of product conception, through development, employment and final retirement. Software projects involve a combination of managers, application specialists, analysts, programmers, and customers. This leads to the need for effective management and communication throughout the entire life cycle in order to achieve the best economically possible product.

One model that generically describes the various processes encompassed by CASE is the hierarchical spiral model [15]. It is based on three levels of modeling for an IS Software product:

- (1) Organizational, at which the organizational role and context of the product is defined
- (2) Conceptual-infological, at which an implementation independent specification for the product is defined,
- (3) Datalogical-technical, at which the technical implementation for the product is defined.

These levels of modeling are structure, function and behavior abstractions from the host organization, universe of discourse, and technology. They match what is now described as upper, middle and lower CASE. ICASE describes a product that supports and integrates various software tools from all three levels of the model.

There are currently several alternate standards for CASE, the most prominent are AD/Cycle in the IBM world and PCTE (portable common tool environment) which is being developed by the European Computer Manufacturer's Association. Both of these standards are still in an evolutionary stage. Market wise the trend is to provide "standalone tools" such as program generators with a limited degree of integration between them. In the long term the direction will be towards supporting the full ICASE environment including enterprise modeling, project management, process management, and reusability. The process of integrating these standalone tools has been described as "an evolution" from a disjointed collection of tools, to a totally seamless environment spanning the complete software development life cycle [3, 8, 27]. More work still needs to be done in order for CASE to span the complete software life cycle. Some foresee that in the final analysis it should be possible to automate the complete SDLC process "the ultimate goal of software engineering will be obtained when a well-thought-out description of a business application can be turned into a running computer program at the push of a button" [38].

## LITERATURE ON CASE

Published articles state that the proper use of CASE

tools to support the process of systems analysis and design can improve both the productivity and quality of work produced in all phases of the systems development life cycle. Therefore, the use of CASE tools has the potential to reduce the problems associated with the manual development of documentation (both written and diagrammatic), requirements definition, business manager/IS professional communication, backlog and maintenance problems experienced in the MIS field [3, 9, 10, 24, 28, 29, 36].

A number of surveys have been conducted on the use of CASE technologies. Based on their results one could attest to the steady increase in the use of CASE, from 4% in 1987 to 24% in 1989 to 30% in 1991 [11, 26, 28]. In contrast, it has been found that "few US shops are using CASE to manage and implement changes to a significant portion of their application portfolios" [25], and potential users have been cautioned to "not plunge" into the CASE adoption process [13, 19, 22]. This study attempts to provide preliminary insight into the reasons behind these findings; specifically the inhibitors and facilitators to CASE adoption.

## METHODOLOGY

### Questionnaire Development

A questionnaire was developed based on issues identified in previous research. Table 1, presents a listing of the categories of issues/benefits synthesized from the corresponding citations. Table 3, includes a listing of the benefits within each category. These were the items included in the questionnaire. Responses were captured on a 7-point scale with poles of "strongly disagree" (1) and "strongly agree" (7). For each item, perceptions were solicited both "prior to implementation" and "after implementation" of CASE.

### The Sample

The sample consisted of 305 organizations located in the States of South Carolina and Georgia. Fifty three organizations fully filled the questionnaire. Of the 53 organizations, 7 were in the service sector, 11 were in the wholesale/retail sector, 13 were in manufacturing, 5 were in mining and 17 were state government agencies. The sizes of the organizations varied from a minimum of 25 employees to a maximum of 1,200 employees, and the size of the IS staff varied from a minimum of 1 employee to a maximum of 70 employees. The individual targeted to receive the questionnaire was either the Director of Information Systems or the Data Processing Manager.

### Method

The responding organizations were classified according to their CASE usage, a summary of which is presented in

**Table 1**  
**Benefits of CASE**

Issue	References
Documentation	[3, 8, 9, 10, 25, 28, 30]
Reliability	[3, 9, 10, 24, 28]
Development/ Methodology	[5, 8, 9, 10, 25, 28]
Productivity	[3, 5, 8, 9, 10, 11, 24, 29, 30]
Training	[5, 22, 29, 34]
Project Control	[5, 24]
Communications	[9, 10, 28]
Visibility	[8, 29]
Financial	[5, 29]

Table 2. The three major groups were: Group 1, which consisted of organizations that had never used CASE; Group 2, which consisted of organizations that had used CASE and then abandoned its usage; Group 3, which consisted of organizations that had used and still continue to use CASE. Organizations that had never used CASE (group 1), were asked to list their perceptions of CASE. Both the organizations who had used and abandoned CASE (group 2) and the organizations that continue to use CASE (group 3) were asked to list their beliefs/perceptions of CASE benefits both prior to and post usage. To facilitate discussion of the results, the terms group 1, 2 and group 3 will be used.

Further follow up discussions were conducted with five of the participating organizations (2 state agencies, 2 from the service industry and 1 from manufacturing). The discussions were conducted with the Director of Information Systems and/or an assigned CASE "expert". The discussions lasted for a period of about two hours and were centered around the questionnaire. The discussions were sufficiently unstructured to allow the participants to present and/or address concepts that had not been included or identified in the questionnaire

## RESULTS

The survey results (Table 2) indicate that 26.4% of the organizations had used CASE, and that a total of 16.9% of the organizations are still using CASE.

The results of the survey are summarized in Table 3. Although the respondent sample size was too small to conduct meaningful statistical analysis, it was appropriate to provide general insight into perceptions of CASE. Our

**Table 2**  
**Respondent's Classification**

Respondent Category	Number	Percentage
Never used CASE	39	73.5
Had used CASE	14	26.4
Used and Abandoned	5	9.4
Continue Use	9	16.9

analysis was thus based on changes in expectation scores from prior to post implementation. Organizations who had never used CASE were asked to state their beliefs about CASE which were then compared to the prior beliefs of the other two groups. To ensure consistency, the numbers were interpreted within "reasonable parameters". These are described below:

1. A seven-point scale was used on the questionnaire. A response of 1 or 2 indicates strong disagreement with the questionnaire statements. A response of 4 indicates marginal agreement and 6 or 7 indicated total agreement. Responses of 3 or 5 are considered to be transitional.
2. During analysis a prior to post difference in expectations rating less than .5 is considered insignificant, a difference of between .6 and 1 is considered as a shift in beliefs and a difference greater than 1 are considered to be a strong shift in beliefs. The existence of variance between the prior and post expectations suggests that the respondents did recognize the scale difference.
3. An increase in prior to post score indicates that after implementation CASE surpassed the a-priori expectations. A decrease indicates that after implementation there was a realization that the benefit was not attainable as expected.

By examining Table 3 we can draw a number of conclusions for each of our three organizational groups (group 1: never used; group 2: used CASE and abandoned; group 3: continue to use). The benefits examined are: documentation, reliability, development-methodology, training, project control, communications, visibility, and financial.

### Benefit Analysis

The survey results show that all of our respondents agree that CASE improves the quality of documentation. The prior expectations for the three groups were similar and there was a positive change for both groups 2 and 3. All groups felt that CASE improves project standards, particu-

**Table 3**  
**Mean Questionnaire Responses**

Issue/Statement	Never Used	Abandoned Use		Continue to Use	
		Prior	Post	Prior	Post
<b>DOCUMENTATION</b>					
CASE improves quality of documentation	4.7	4.6	5.2	4.7	5.3
Using CASE improves project standards	5.3	4.4	4.6	5.2	6.0
CASE makes development of graphical representations easier	4.8	3.8	5.0	4.9	5.8
<b>RELIABILITY</b>					
CASE results in products of higher reliability	4.7	5.0	5.0	5.3	5.0
Using CASE, more extensive analysis and design is likely to occur prior to coding	5.6	5.2	6.4	5.8	6.2
Using CASE results in less complex code	4.3	4.4	4.4	4.6	4.6
CASE results in products that are easier to test	3.9	4.2	4.8	4.8	5.1
<b>DEVELOPMENT/METHODOLOGY</b>					
CASE replaces existing methodologies or techniques	4.1	4.8	3.0	5.4	5.3
CASE is used for most new software development	3.3	5.2	5.0	4.7	5.2
Maintaining CASE developed software is easier	3.8	4.6	5.4	5.0	6.3
It is easier to re-develop an application using CASE than to maintain an application developed prior to CASE	4.6	3.0	3.0	4.3	4.2
It is easier to develop a system using CASE	4.7	5.4	4.6	5.1	5.7
<b>PRODUCTIVITY</b>					
CASE increases programmer productivity	4.8	5.6	4.4	4.8	5.0
CASE is perceived as a threat by technical personnel	2.7	2.8	2.0	4.7	4.0
CASE projects are completed on time	3.8	5.2	4.6	4.6	6.0
Using CASE will substantially reduce the IS applications development backlog	4.2	3.0	4.0	4.8	4.9
<b>TRAINING</b>					
CASE takes longer to learn than other software tools	3.2	3.0	3.4	3.6	4.2
Training is more critical to the successful implementation of CASE, as compared with other software tools	3.8	5.6	5.8	5.1	5.9
<b>Project Control</b>					
Using CASE it is easier to track the progress of products as they are being developed	4.5	3.8	5.6	4.8	5.9
Using CASE it is easier to control product development	4.6	3.8	5.2	4.9	5.7

Table 3 (continued)

Issue/Statement	Never Used	Abandoned Use		Continue to Use	
		Prior	Post	Prior	Post
<b>COMMUNICATIONS</b>					
CASE improves communications between technical and user personnel	4.7	5.2	6.6	5.0	5.7
CASE improves communications among technical personnel	4.6	5.8	6.0	5.3	5.2
CASE improves the chances that the developed systems meet user needs	3.4	4.6	5.6	4.8	5.4
<b>VISIBILITY</b>					
CASE improves IS's recognition by management	5.3	4.8	5.0	4.4	4.7
CASE benefits are immediately visible	3.2	1.0	3.0	3.8	4.6
CASE benefits are only visible in the long term	4.3	3.0	3.0	4.6	5.1
<b>FINANCIAL</b>					
In the short term using CASE saves money	2.7	4.8	5.6	3.3	3.6
In the long term using CASE saves money	4.7	5.6	5.6	5.9	5.9

larly group 1, while group 3 showed a positive shift after usage. All three groups concurred on the ease of development of graphical representations using CASE with both groups 2 and 3 showing a strong positive shift in beliefs after usage.

### Reliability

All respondents believed that CASE resulted in products of higher reliability, with no significant change in their beliefs prior and post implementation for both groups 2 and 3. Although all groups believed that using CASE more extensive analysis and design is liable to occur prior to coding. In fact the group that abandoned CASE usage showed the most significant prior to post change. This could indicate that they spent a longer time designing products than those who continue to use CASE. There is general agreement that using CASE results in less complex code and products that are easier to test though no group changed their beliefs after implementation. Group 2 showed a positive change in their belief that CASE developed products are easier to test. This could be attributed to a number of factors such as lower initial expectations than organizations who continue to use CASE, incomplete testing, as a result of having abandoned the CASE developed products prior to implementation, no large projects implementations, etc.

### Development/Methodology

While all groups initially believed that CASE replaces existing methodologies group 2 showed a large negative shift in beliefs after implementation. They also indicated a negative shift regarding ease of development using CASE. This provides some indication for the reasons for abandonment by this group. They did however, indicate a positive shift in beliefs regarding the ease of maintenance of CASE developed software. Group 3 indicated positive shifts for new software development, ease of development, and a strong positive change for the ease of maintenance. Interestingly, Group 1 believed that it is easier to re-develop an application using CASE than to maintain an application developed prior to CASE. Group 2 disagreed with the statement, and Group 3 marginally agreed with the statement.

### Productivity

All three groups initially expected CASE to increase programmer productivity. Group 2 showed a strong negative shift in beliefs indicating that they did not succeed in achieving their a-priori expectations. Organizations that had not implemented or had abandoned CASE implementation indicated that it was not viewed as a threat by technical personnel, while the organizations that succeeded stated that it was viewed as a threat by technical personnel. Both sets of organizations that tried CASE showed that after using CASE

they perceived it to be less of a threat than originally anticipated. There was a sharp contrast between Groups 2 and 3 on whether their projects were completed on time. Group 2 tended to disagree with this after implementation while Group 3 showed a strong positive shift in beliefs. However, Group 2 did show a positive shift in beliefs regarding the reduction of application backlog through CASE.

### **Training**

Group 1 organizations believed that CASE is no different in terms of training time or complexity than any other software tool. Group 2 organizations believed that it did not take longer to learn than other software tools, but that it was more critical to the successful implementation of the tool. Group 3 organizations did not initially agree that CASE takes longer to learn, but after implementation showed a positive shift and marginally agreed. They did believe that training was more critical to successful implementation as compared with other software tools, and showed a positive change in their beliefs.

### **Project Control**

All respondents agreed that it was easier to track and control product development, with strong positive shifts between the prior and post beliefs.

### **Communications**

There was general agreement that communication between the users and technical personnel was improved, with a strong positive shift indicating that it exceeded prior expectations. There was also strong agreement that CASE improved communications between technical personnel. Both groups 1 and 2 agreed that CASE improved the chances of meeting user needs and indicated a positive experience after implementation.

All organizations agreed that CASE improves IS's recognition by management. Both Group 2 and Group 3 organizations experienced positive shifts in attitude toward CASE's short term benefits. Only Group 3 organizations however, who continued to use CASE, tended to agree more with the statement that CASE's benefits are only visible in the long term.

### **Financially**

Differences in projected costs between short and long term were identified by the three groups. Group 1 organizations showed little belief in short term money savings, but did believe in the long term money savings. Group 2 organizations believed in both the short and long term money savings of CASE both prior to implementation and post abandonment, and even showed a positive change in beliefs

regarding the short term savings. Group 3 organizations primarily believed in long term money savings with no change in their prior to post implementation beliefs.

## **DISCUSSION**

The results in Table 3 are now discussed in terms of the three groups. While many interpretations can be made directly from the numbers in the table, others are somewhat speculative.

### **Group 1: Organizations That Never Used CASE**

This group believes in most of the proclaimed benefits of CASE such as: improved documentation, improved reliability, easier system development, increased programmer productivity, increased ease in tracking and controlling product development, improved communications, and increased IS recognition by management. So, why have they not implemented CASE? Some reasons can be inferred from their disagreement with the fact that CASE can be used for most new software development and it improves chances of developed systems meeting user needs. In other words, this group while acknowledging the quality and control issues of system development through CASE, question its widespread applicability in meeting user needs.

Perhaps more importantly, this group does not see CASE benefits visible in the short run. While this is not necessarily true in all organizations, our discussions indicated that these organizations are deficient in one or more of the following areas.

- (a) They do not possess the necessary hardware base on which to implement CASE.
- (b) They do not have experienced personnel capable of implementing CASE.
- (c) They may not have reached the integration stage of growth in which they would need to use sophisticated products (such as CASE). Many of their software products remain standalone, batch programs, or off the shelf, in which case they would not need CASE.

### **Group 2: Organizations that Used and Abandoned CASE**

This group also believes in most of the potential benefits of CASE. The positive belief changes that they displayed were that CASE improves the quality of documentation (written and graphical), allows for more extensive analysis prior to coding, results in products that are easier to test and maintain, will reduce the IS application backlog, makes it easier to track and control product development, improves communication, and saves money in the short term. So,

why did they abandon CASE? These organizations displayed some negative shifts in belief with respect to the following statements: CASE replaces existing methodologies or techniques, it is easier to develop systems using CASE, CASE increases programmer productivity and CASE projects are completed on time. This would suggest that they found CASE difficult and slower to use than previously expected.

While this is not necessarily true in all organizations, our discussions indicated that these organizations had problems in one or more of the following areas: (a) Lack of management support. (b) Lack of user support. (c) Lack of training. (d) Indulgence in the technology rather than the application. (e) Organizational change.

### Group 3: Organizations that Continue to Use CASE

This group believed in all of the proclaimed benefits of CASE. The positive belief changes displayed by this group were improved documentation (both written and graphical), improved project standards, used for most new software development, easier to develop and maintain systems using CASE, projects completed on time, training is more critical, easier to track and control project development, improves communications and the chances that systems developed meet user needs, CASE benefits are visible immediately and in the long term.

### Barriers to Adoption and Implementation

The barriers to CASE may be divided into two sets. The first set of barriers are those that prevent an organization from adopting CASE, while the second set of barriers prevent an organization from succeeding in its CASE implementation. The results discussed in the above sections provide a number of insights into the barriers to CASE adoption and implementation. These insights are discussed below and are based on both the questionnaire data and the in-depth interviews.

#### Barriers to Adoption

The barriers to adoption are based on perceptions by both the IS department and company management. These barriers can be roughly gauged by comparing the beliefs of group 1 to the prior beliefs of groups 2 and 3. The barriers identified are:

- (a) **Perceived complexity** or the perception that CASE is a complex product that results in products that are difficult to test and maintain. It is also difficult to use for all new software development.
- (b) **Inability to meet user needs** is perhaps a critical perception that inhibits adoption. In other words, Group 1 organizations do not believe that CASE improves the chances that the developed systems

meet user needs.

- (c) **Low payoff** of CASE is a potential barrier to adoption as Group 1 organizations do not believe that CASE saves money in the short term, and marginally believe that it will save money in the long term.

Therefore it seems that non-adoption is partially a function of a-priori beliefs regarding CASE. Adoption will be facilitated through education on the product itself and its financial benefits.

### Barriers to Implementation

The barriers to implementation are those that prevent an organization from implementing CASE on a permanent basis. They are identified in the survey through the differences between groups 2 pre to post implementation beliefs. These indicated that organizations that abandoned CASE did not feel they could use it to replace existing techniques, had difficulty in actually implementing systems through CASE, experienced no increase in programmer productivity, could not complete CASE projects on time, and saw little or no short term benefits of CASE.

The interviews revealed that these negative belief shifts are based on four fundamental barriers to successful implementation: lack of training, inability to adopt to change, selection of inappropriate CASE tools and lack of top management support. These are discussed in depth below.

#### Lack of training

The amount, type and scope of training required is a function of the level of CASE implementation. Lower CASE requires mainly technical training, structure and conformity are enforced on the programming teams. Upper CASE requires training in communication, procedures, operational practices and business process design, training in new thought processes, linking the IS plan and the business plan. Middle CASE requires a mixture of both.

This has also been identified in other research. For example [34] states that "CASE is a process for which most data processing professionals have no training". La Plante [22] states that "One explanation for the growth of CASE within organizations being less than anticipated is improper training. One reason CASE training lags behind other subjects is that it is more difficult. According to CASE experts there are some precepts of progressive CASE training. First, CASE methodology and techniques must be taught in addition to CASE tools being used in a specific project. Second, CASE training must be customized to fit each corporations unique needs".

This underestimation of training needs causes initial

CASE projects to take more time than if they were done using current techniques. In our survey we found that both groups 1 and 2 underestimated the time taken to learn CASE. While group 3 organizations initially believed that CASE training took longer and during implementation experienced a positive shift in belief. They also displayed a stronger shift in belief regarding training's criticality to the successful implementation of CASE. This is true even in the simplest CASE adoptions (lower CASE). Time must be set aside specifically to train involved personnel. Restated, the time taken to learn CASE added to the project time using CASE is always greater than the time taken using current methods. It is only after CASE has been learned that the project time drops and as the repository grows and organizational issues are overcome the project time continues to drop.

#### **Inability to adapt to organizational and cultural change**

This is a major problem. It is a result of changes that are more procedural/organizational than technical. The higher the level of the CASE implementation the more widespread and complex the problem becomes. In the survey, the slight improvement in recognition of IS by management would tend to indicate that most of the CASE implementations were either lower or middle CASE, and that the organization wide impact was minimal, i.e., the organization was exposed to more of the software development cycle (as indicated by the improved communication with users response) but did not have to change its method of conducting business, either organizationally or culturally, as a result of the CASE implementation. This problem has been previously discussed in the literature [23, 39], other examples can be drawn from Sharon [34], who suggests that "In implementing CASE technology transfer, MIS managers must make positive change occur frequently — and in manageable doses", and Holcman [14] states that "the magnitude and importance of this change demand that systems development organizations understand and manage the cultural and organizational changes that will take place". Knight [19], discusses this problem in terms of a supporting methodology and states that "CASE tools can only provide minimal benefits when they are implemented without a supporting methodology".

#### **Selection of an inappropriate tool or set of CASE tools**

This problem was identified in almost all the follow up discussions conducted with five of the participating organizations, and may be summarized as follows. All CASE products (like all other software products) have their nuances, some things are easier to accomplish than other, a particular method for doing something is easier to do using one set of software versus another. This will in turn impose restrictions on the programmer/analyst, and he/she, must work within those restrictions. In the literature this has been discussed by Foley

[7], who states that despite success stories, CASE continues to be burdened by: (a) a lack of standards, (b) vendors that support only one platform, methodology, and/or piece of the complete software development life cycle, (c) primitive dictionary-repository facilities, (d) little if any integration among tools. This problem will be gradually reduced over time as CASE products get closer to maturity.

#### **Lack of long term management commitment, support and understanding**

This has always been a problem with IS projects, and will remain a problem in lower and middle CASE implementation, whether this can be overcome through Upper CASE implementation is yet to be determined, as there appear to be few organizations that have truly implemented Upper CASE, and the outcome of these implementations have not yet been analyzed in the research literature.

### **CRITICAL SUCCESS FACTORS**

The successful implementation of CASE technology requires planning, support, and control from management, [4, 24]. Based on the above discussion, there are certain factors critical to the success of the implementation that must be watched closely:

- (a) **Integration;** both integration among the CASE tools, and integration of CASE into the work flow.
- (b) **User involvement;** from the beginning a large amount of ownership must be transferred to the users.
- (c) **Definition of standards;** these need to be set and strictly adhered to.
- (d) **Proactive support from management** to the IS group and from the IS group to the users.
- (e) **Progressive implementation;** projects need to be implemented in small progressively larger successful steps.
- (f) **Continuous training** for all those involved in the CASE implementation.
- (g) **Managing change;** preparing for, monitoring and controlling the organizational changes that will take place.

### **CONCLUDING REMARKS**

The survey results indicate that most respondents do not use CASE. Some organizations have experimented with the technology and have been unable to implement it. Only a small percentage of all organizations have implemented CASE on a permanent basis in their organization. As the CASE tools improve and standardize (which they are rapidly doing today) some of the problems of implementing CASE



will disappear and more organizations will implement it. Today we seem to be past the stage of early adopters, with the appearance of CASE and its standardization, CASE will begin moving into many IS shops, eventually most if not all organizations will use CASE (at some implementation level).

The major benefits stem from improved documentation, improved communication, improved work processes, and reusability at all levels. Some benefits such as reusability can only be achieved over long periods of time (as the repository grows). The most important benefit, which is yet to be measured, is the competitive and strategic advantages that may be realized through the increased focus on data, processes and information at an organizational level.

The major problems are lack of training, dealing with the organizational/cultural changes imposed, a current lack of standards, and lack of management support/involvement. Generally speaking the larger the implementation (both horizontally and vertically) the greater the gains (result of synergy), and the greater the risks.

During initial implementation one must be able to measure productivity and compare prior to post CASE. In selecting projects it is best to develop new projects using CASE and allow old systems to die of obsolescence rather than to try and replace them with CASE developed systems. Also, companies that have taken time to learn CASE and to readjust their culture have significantly gained from CASE.

In sum, the study provides both MIS practitioners and academicians with a view of the current status of CASE. Practitioners are offered a survey of the CASE adoption/implementation process, including when and how to implement CASE, and which factors are critical during the implementation process.

The survey and interviews in this study represent an exploratory methodology aimed at gaining insight into the use (or non-use) of CASE in American organizations. The small sample size, possible non-response bias, and lack of statistical analysis of findings raise questions of external validity of the survey. A number of propositions are implicit in this paper that need to be validated using larger sample sizes and more rigorous methodologies. For example, a possible model implicit in this study relates the benefits of CASE to several factors; (a) the level of CASE implementation (upper, middle, lower CASE, or ICASE), (b) the extent of CASE implementation, both horizontally (number of individuals involved, programmers, analysts, etc.), and vertically (management, organizational involvement), (c) the degree of involvement (duration of use, number of projects implemented, project size, project complexity), (d) the selected tool(s) and its limitations. This and similar models can be tested in further research so that prescriptions regarding the use of CASE tools can emerge. MIS researchers are also presented with some validation of a pattern being followed

in the adoption of CASE technology, and an analysis of how the barriers to both implementation and success affect the growth of the technology. Future research could follow up the investigation of CASE adoption and pattern development. Specific areas of consideration would be how prepared the organization was to adopt CASE, how CASE can increase competitive advantage by re-focusing the organizations views on data, processes and information, and whether this type of research is extendible to other emerging software technologies.

## REFERENCES

- [1] Apte, Uday M., Sankar, Chetan S., Thakur, Meru, and Turner, Joel E., "Reusability-Based Strategy for Development of Information Systems: Implementation Experience of a Bank", *MIS Quarterly*, Vol. 14, Dec. 1990, pp. 421-433.
- [2] Bauer, F.L., "Software Engineering", *Information Processing*, North Holland Publishing Company, 1972, pp. 530.
- [3] Burkhard, Donald L., "Implementing CASE Tools", *Journal of Systems Management*, May 1989, pp. 20-25.
- [4] Brisebois, Ronald; Dion, Pierre; "Implementing CASE: Six Critical Factors", *Computing Canada*, Vol. 16, Num. 11, May 24, 1990, pp. 21-36.
- [5] Danziger, Michael R. and Haynes, Phillip S., "Managing the CASE Environment", *Journal of Systems Management*, May 1989, pp. 29-32.
- [6] Fisher, Alan S., "CASE: Using Software Developmental Tools", John Wiley & Sons, Inc., 1988.
- [7] Foley, Mary Jo, "Agencies Go Slow in Adopting CASE", *Systems Integration*, Vol. 23, Num. 9, Sep. 1990, pp. 77-94.
- [8] Gibson, Michael L., Snyder, Charles A. and Kelly, Ranier R.Jr., "CASE: Clarifying Common Misconceptions", *Journal of Systems Management*, May 1989, pp. 12-19.
- [9] Gibson, Michael L., Snyder, Charles A. and Houston, H., "Implementing a Corporatewide Information Strategy Through CASE", *Journal of Systems Management*, May 1990, pp. 8-17.
- [10] Gibson, Michael L., Snyder, Charles A. and Houston, H., "Why CASE Belongs in Strategic Business Management", *Information Strategy: The Executive's Journal*, Vol. 7, Num. 2, Winter 1991, pp. 17-23.
- [11] Graham, Carol, "Case Cracks Applications Backlog", *Datamation*, Vol. 37, Issue 6, May 15, 1991, pp. 97-99.
- [12] Hanson, S.J., and Rosinski, R.R., "Programmer Perceptions of Productivity and Programming Tools", *Communications of the ACM*, Feb 1985, pp 180-189.
- [13] Hayley, Kathryn J. and Thaine H. Lyman, "The Realities CASE", *Journal of Information Systems Manage-*

- ment, Vol. 7, Num. 3, Summer 1990. pp. 18-23.
- [14] Holcman, Samuel B., "Preparing for AD/Cycle", *Computerworld*, Nov. 26, 1990, pp. 69-76.
- [15] Iivari, J., "Hierarchical Spiral Model for Information System and Software Development. Part 1: Theoretical Background", *Information and Software Technology*, Vol. 32, Num. 6, Jul/Aug. 1990, pp. 386-399.
- [16] Katz, R.L., "Business Enterprise Modelling", *IBM Systems Journal*, Dec 1990, pp. 509-526.
- [17] Kernerer, C.F., "An Agenda for Research in Managerial Evaluation of Computer Aided Software Engineering [CASE] Tool Impacts", *Proceedings of the Twenty Second Annual Hawaii International Conference on System Sciences*, Kona, Hawaii, January 3-6, 1989.
- [18] Keuffel, W., "CASE for the rest of Us", *Computer Language*, Jan 1991, pp. 25-29.
- [19] Knight, Robert, "Don't Plunge into CASE", *Software Magazine*, Vol. 10, Num. 9, July 1990, pp. 28-30.
- [20] Knight, Robert, "Shop Floor Accepts CASE at a Caterpillar's Pace", *Software Magazine*, Vol. 10, Num. 3, March 1990, pp. 72-73.
- [21] Kozaczynski, W., "A Suzuki Class in Software Engineering", *IEEE Software*, Jan 1991, pp. 97-99.
- [22] LaPlante, Alice, "Making a CASE for Better Training", *Computerworld*, Feb. 25, 1991, pp. 100.
- [23] Lederer, Albert L., "Making Strategic Information Systems Happen", *Academy of Management Executive*, Vol. 4, Num. 3, August 1990, pp. 76-83.
- [24] Lin, Chan-Yang and Chen-Hua Chung, "End-User Computing in a CASE Environment", *Journal of Systems Management*, Vol. 8, Spring 1991, pp. 17-21.
- [25] Maria, Arturo G., "CASE Technology: Today's Reality", *Journal of Systems Management*, Feb 1991, pp. 18-20.
- [26] McLure, C., "Computer Aided Software Engineering", *Computer Aided Software Engineering*, Digital Consulting, Andover, MA, Spring 1987.
- [27] McLure, C., "Case in Software Automation", Englewood Cliffs, New Jersey: Prentice-Hall, 1989.
- [28] Necco, Charles R., Tsai, Nancy W. and Kreg W. Holgeson, "Current Usage of Case Software", *Journal of Systems Management*, May 1989, pp. 6-11.
- [29] Norman, Ronald J., Corbitt, Gail F. Butler, Mark C. and Donna D. McElroy, "CASE Technology Transfer: A Case Study of Unsuccessful Change", *Journal of Systems Management*, May 1989, pp. 33-37.
- [30] Norman, Ronald J., Nunamaker, J.F. Jr, "CASE Productivity Perceptions of Software Engineering Professionals", *Communications of the ACM*, Sept. 1989, pp. 1102-1109.
- [31] Oman, P.W., Bowles, A.J., Mount, R., Karem, G., Kalinsky, D., Tervonen, M., Bundonis, V., Fischer, H., Fish, M., Longshore, D. and Akiha, N., "CASE Analysis and Design Tools", *IEEE Software*, May 1990, pp. 37-42.
- [32] Poulin, J.F., "Object-Oriented CAD Database Support for Software Reusability in Computer-Aided Software Engineering Environments", *Unpublished PhD Thesis*, Rensselaer Polytechnic Institute, 1989.
- [33] Protsko, L.B., Sorenson, P.G., Tremblay, J.P. and Schaefer, A.D., "Towards the Automatic Generation of Software Diagrams", *IEEE Transactions on Software Engineering*, Jan. 1991, pp. 10-22.
- [34] Sharon, Dave, "Negotiating the CASE Trails", *Software Magazine*, January 1991, pp. 107-108.
- [35] Stamps, D., "CASE Cranking out Productivity", *Datamation*, July 1, 1987, pp. 55-58.
- [36] Suydam, W., "CASE Makes Strides Towards Automated Software Development", *Computer Design*, January 1987; pp. 49-51, 58.
- [37] Taylor, J., "Information Systems and Structured Techniques", *Industrial Management & Data Systems*, July 1990, pp. 9-15.
- [38] Williamson, Mickey, "Getting a Handle on CASE", *CIO*, Vol. 3, Num. 7, Apr 1990, pp. 42-51.
- [39] Zagorsky, Carol, "Case Study: Managing the Change to CASE", *Journal of Information Systems Management*, Vol. 7, Num. 3, Summer 1990.

#### ABOUT THE AUTHORS

*Varun Grover is an Assistant Professor of MIS at the University of South Carolina. He received his Ph.D. degree from the University of Pittsburgh. His current research interests include the management and development of IS and telecommunications systems. His articles have appeared or are forthcoming in Journal of Management Information Systems, Information and Management, Decision Sciences, Interfaces, IEEE Transactions in Engineering Management, Database, Omega, Journal of Information Systems Management, and numerous other publications.*

*Hassan Elshazly holds a Ph.D. degree from the University of South Carolina. He is currently the Director of Information Systems at the Department of Youth Services, a unit of the State of South Carolina. He has over 15 years of experience in various MIS positions in the industry. His current research interests are in the areas of systems development and evaluation.*