University of Northern Iowa

UNI ScholarWorks

Presidential Scholars Theses (1990 – 2006) Student Work

1995

Critical success factors for organizational CASE success

Chad Strauss

University of Northern Iowa

Let us know how access to this document benefits you

Copyright ©1995 Chad Strauss

Follow this and additional works at: https://scholarworks.uni.edu/pst

Part of the Management Information Systems Commons, and the Software Engineering Commons

Recommended Citation

Strauss, Chad, "Critical success factors for organizational CASE success" (1995). *Presidential Scholars Theses (1990 – 2006)*. 144.
https://scholarworks.uni.edu/pst/144

This Open Access Presidential Scholars Thesis is brought to you for free and open access by the Student Work at UNI ScholarWorks. It has been accepted for inclusion in Presidential Scholars Theses (1990 – 2006) by an authorized administrator of UNI ScholarWorks. For more information, please contact scholarworks@uni.edu.

Offensive Materials Statement: Materials located in UNI ScholarWorks come from a broad range of sources and time periods. Some of these materials may contain offensive stereotypes, ideas, visuals, or language.

Critical Success Factors for Organizational CASE Success

Chad Strauss
University of Northern Iowa

Abstract

Practitioners and researchers have suggested that Computer Aided Software Engineering (CASE) technology can significantly improve the productivity of systems analysts and the quality of systems development. Before CASE can succeed, however, specific components must exist. This paper explores CASE and the components necessary for its widespread acceptance and use within information systems departments. Specifically, this paper will examine the systems development life cycle (SDLC), the possible benefits of CASE, the current state of CASE, critical success factors for CASE, and finally CASE use.

Introduction

With the ever increasing competition in todays markets, firms are continually looking for the competitive advantage necessary to position themselves as the leader in their industry. One integral component of being the premier firm is the use and management of information. It is often the company that manages its information the most effectively and efficiently that maintains the number one position. The cost, however, of information technologies is high and continues to grow. As a result, businesses are looking for opportunities to squeeze more out of their information systems dollars. One alternative is the exploration of computer aided systems engineering (CASE) tools.

"CASE is viewed as a strategy to reduce development time, to cut maintenance costs, and to improve the discipline of information systems development" (Sumner and Ryan 1994, p.16). To do this CASE sets out to automate the current manual process of systems development, with the ultimate goal of ". . . enhancing the quality and reliability of systems" (Jones 1992, p. 38). Initially CASE technology was viewed as a panacea, the long awaited savior for system developers. While these benefits can and have been realized by organizations, it is often a long and difficult road to success.

Before embarking on the examination of CASE, some background information concerning information systems development is necessary to build a foundation upon which CASE technology is built. This foundation is best explained through an analysis of the process followed by systems analysts to develop information systems. This process is generically known as the

Systems Development Life Cycle (SDLC).

For successful completion of a project, regardless of the discipline under study, a careful and thoughtful plan to follow is imperative. Just as a student follows a plan to write a paper (research, outline, interview, rough draft, etc.), so does a systems analyst when designing a new information system. This methodology varies from organization to organization, but the essential components are always present. Our SDLC will include the following processes:

1) Systems Planning
2) Systems Analysis
3) Systems Design
4) Systems Implementation
5) Systems Support

Specifically, a systems development life cycle is "a process by which systems analysts, software engineers, programmers, and end-users build information systems and computer applications" (Whitten 1994, p.91). It is a management tool used to plan, execute, and control systems development projects.

Systems Planning

This phase has an organizational wide scope and is designed to identify and prioritize information systems applications whose development would benefit the entire organization.

Systems Analysis

Analysis focuses on a single application chosen from the planning phase. It's purpose is to analyze the business problem and define the business requirements to improve or create a new information system.

Systems Design

Having the requirements now defined, analysts set out to design a computer-based, technical solution to meet the business requirements previously defined. This includes designing program specifications, controls, security, etc.

Systems Implementation

Here the purpose is to actually construct the technical components of the system and deliver it into operation. This phase tends to receive the most attention, because it tends to be more tangible in nature. It is imperative, however, that the previous phases be thoroughly executed. In fact the previous phases are often more critical than implementation. If proper analysis and design are not done, implementation can not possibly succeed.

Systems Support

Finally, the purpose of support is to sustain and maintain the system for the rest of it's useful life. This means providing user support, documentation, and necessary modifications (Whitten 1994, pp. 101-102).

This five phase process is the backbone for developing information systems. So, where does CASE technology fit into this? First of all, CASE is not a new SDLC. In fact it isn't a life cycle at all. CASE can best be viewed as a toolbox that analysts use to help them through the life cycle. Just as a carpenter would use a blueprint to build a house, he or she would use a hammer to do the actual work. For a systems analyst, he or she would use a life cycle methodology to build an information system, but would use CASE tools to do the actual work.

CASE technology is not all that new. The tools have been around for at least a decade, but only recently have they begun to receive significant attention, primarily due to the fact that their capabilities have greatly been enhanced and the hardware prices necessary to operate them have fallen. In some ways CASE is a descendant of computer aided design (CAD) tools that engineers use. CAD, like CASE, are tools used to automate the process of engineering, whether it be electrical or information engineering. The purpose, as alluded to, is to assist analysts to design higher quality systems, in less time, and with less effort.

The impacts of CASE on the SDLC are crucial. CASE tools can be divided into upper-CASE tools and lower-CASE tools. Upper-CASE tools tend to support planning, analysis and general systems design, while lower-CASE tools refer to detailed systems systems design, implementation, and support. Additionally, cross life cycle CASE refers to tools that support the entire life cycle, such as project management and feasibility assessments. Essentially, CASE is a broad group of software technologies that together support the automation of systems development and can help reduce the programming backlog that has long existed (Steinberg, 1992).

This "toolkit" of upper and lower CASE tools are designed to work together to support a phase (or preferably the entire cycle) of the SDLC. The tools, ideally, would be integrated (I-CASE) so that specifications are passed from one stage of the development cycle to the next. Examples of such tools (roughly fitting the SDLC phases) include:

1) analysis tools that aid in diagraming and prototyping, including a specification checker and a database for saving specifications;

2) data design tools that perform conversion of logical data models to design specifications and the automatic generation of database schemes;

3) programming tools that generate compilabe code from design specifications;

4) maintenance tools that assist in assessing the system impact of a maintenance request, and also provide reverse engineering capabilities and;

5) project management tools that track an help managers control projects (Steinberg, 1992).

While CASE technology sounds like a major breakthrough in the art and science of information systems engineering, it has encountered its share of problems. Some are technical in nature, but many are organizational and people oriented. It is this area that will be of primary concern in this paper. While some technical aspects will be addressed, they will not be stressed. The essential question to be answered, is "if CASE is so great, why have many organizations failed to use it effectively?" Though this question will not be explicitly answered in the paper, it is the central theme and will provide the reader a reference point to draw their own conclusions.

CASE Benefits and Productivity Returns

The benefits promised by CASE can be categorized into four general categories; increased productivity, improved quality, better documentation and reduced lifetime maintenance (Whitten 1994, pp. 195-197).

Increased Productivity

Because CASE is designed to automate many of the tedious activities by analysts, the hope is that CASE can bring about higher levels of productivity, thus reducing development time. Estimates of improved productivity range from 35 to more than 200 percent, though estimation is difficult and the margin of error can be high

(Whitten 1994, p. 195). Productivity, however, takes time as CASE technology has shown to have an enormous learning curve. Because the tools can be highly complex and offer a myriad of features, productivity often declines initially.

Improved Quality

The success of any project can be measured in various ways. Does it fulfill user requirements, can it be easily modified, is it bug-free and so on. CASE can eliminate or significantly reduce errors and omissions assuming the analysts apply a sound methodology. Like a spell-checker, many CASE products can "proof-read" your work, searching for possible problem errors.

Better Documentation

CASE tools can make it easier to maintain documentation. Often documentation is negligible or non-existent simply because developers aren't willing to take the time to document thoroughly. With CASE, the tools necessary to perform accurate and timely documentation are easily accessible and easy to use, therefore allowing for better documentation.

Reduced Lifetime Maintenance

The combination of higher qua4/23/954/23/95lity systems and better documentation should be a reduction in costs and effort required to maintain the system. It has been estimated that over 70 percent of the time analysts spend on the job is spent on maintenance. This includes enhancing current systems, altering systems to comply with new procedures or laws, or correcting bugs and errors. By reducing the amount of time spent on maintenance (because the new systems are of higher quality, better documented, and more easily changed) the analyst can spend additional time reengineering older systems and focus on developing new systems that can return a greater benefit to the organization.

As was previously stated, it often is not the technical aspects of CASE that prevent actualization of these benefits (though it is at times), but rather the organizational and people aspects that prevent full CASE benefits from being utilized. With this understanding, lets explore the current state of CASE technology.

Current State of CASE technology

Technically speaking, CASE technology has matured at a phenomenal rate. Initial CASE tools could perform limited functions such as basic diagraming, rudimentary code generation and simple documentation. In recent years, however, CASE has evolved into a powerful set of tools for systems analysts, though many more improvements are still necessary.

Current use focuses on selected tools, however. An IS department may begin with exploring diagraming tools. Another may begin with utilizing code generators. Other aspects of the SDLC are still performed manually. Many organizations, however, never progress past this use of the technology. In fact studies suggest that only about 24% of companies use some form of CASE tools at all (Steinber, 1992). While their specific use may result in positive benefits, the true value of CASE is never fully utilized.

The organizations that have found significant benefit tend to be those employing integrated-CASE (I-CASE tools). Some vendors are offering complete packages of CASE tools to work together, while others are working on developing standards for the industry so that each component/tool can work with others. What this means is that eventually (and to some extent currently) an analyst can progress throughout the entire SDLC with the use of CASE. Planning and prioritizing of new systems can be done on the computer. Analysis of the business requirements can be translated into diagrams. Specifications can be taken from the diagrams and can be converted into algorithms and database designs, automatically. Code can be generated, tested and installed, all automatically by the CASE tool. The possibilities seem to be endless, though true integration is yet to arrive, and those that offer it still have many shortcomings.

So, what do the users (the analysts) of CASE really think? What tools do they use? What benefits do they perceive? What problems have they encountered? A survey of 400 CASE users found that most analysts used CASE tools associated with analysis and design activities. Among the best benefits and those that were most realized were the easy modifications allowed in preliminary designs, better standardization and easier maintenance. Conversely, a higher quality system that was deemed very important has yet to be realized (Yellen, 1992).

Among the problems noted include the large consumption of computer resources necessary to use the tools, lack of top management support and lack of true integration of system components (Yellen, 1992). Overall, the users surveyed are "somewhat satisfied" with CASE. It is obvious from this study that CASE tools are apparently not used to their full potential. It is clear that CASE tools have been extensively utilized for lower level tasks such as drawing diagrams and descriptions of the system. The more sophisticated tools, such as code generators, are used substantially less. CASE does not appear to help developers create higher quality systems and at higher productivity levels. Many of these issues can be explored as we begin exploration into the organizational issues surrounding CASE technology.

Critical Success Factors for CASE

A: Organizational Context

With a foundation to build upon now established, we turn to examine what is necessary for CASE to succeed in an organization. Not the technical aspects (yet anyway), but rather the more theoretical aspects. Essentially, there are three general segments for exploration, an organizational context an analyst's skills context, and a technical context. Each segment contains many dimensions and variables, that must be oriented in such a way as to promote successful CASE adoption within an organization. In general they are called critical success factors. These critical success factors are defined as "what must go right to achieve successful results" (Sumner and Ryan 1994, p.17).

Organizational issues will be examined first. In other words, "what is necessary in the organization as a whole" for CASE to succeed? Arun Rai, an Assistant Professor of Management Information Systems at Southern Illinois University and Geoffry Howard of Kent State University have studied and researched organizational issues pertaining to CASE technologies extensively, of which will be used and expanded upon greatly here. To begin this analysis, one key assumption must be accepted. That assumption is that CASE is an <u>innovation</u> (Rai and Howard, 1993). Specifically it is an innovation in the field of information systems development. While it is true that CASE is primarily viewed as a technical development in designing information systems, it is really an exploration of organizational innovation. In this light we will depart from the technical details of CASE and investigate the necessary variables that foster an innovation, such as CASE. In other words, because CASE is an innovation we want to explore variables that foster innovation, specifically those that can be correlated to CASE technology.

It is helpful to view organizational innovation as a two stage process consisting of initiation and implementation (Rai and Howard, 1993). Initiation consists of the acquisition and experimentation of the innovation, in this case, CASE technology. It is important to note, however, that this segment of the innovation is a separate phenomenon from its actual routine usage as defined by the implementation phase. Essentially, initiation is brought about by a few individuals that explore various aspects of an innovation. An organizational subunit is generally formed to do this. Note, however, that simply possessing the technology does not assure its usage, except on an experimental or trial basis.

Following initiation, innovations including CASE technology, undergo a transition to implementation. The transition is not abrupt however, it is generally subtle and the exact point is usually blurred.

At this second and last stage the innovation moves from being experimental to diffusing throughout the organization. This two-stage phenomenon process is generally quite evident with most new innovations, such as CASE technology.

This two-stage view of innovation can be easily linked to CASE by viewing CASE as many various aspects. Most CASE products are simply pieces of a whole toolbox or integrated-CASE (I-CASE) array of products. Information systems departments (ISD's) explore each of these tools/aspects by actually acquiring them, testing them, and evaluating them. Experimental groups explore its capabilities, and if found beneficial, proceed to diffuse the technology to the rest of the ISD. Appendix A depicts how various aspects of CASE are acquired, experimented with, and finally diffused throughout the organization.

With this understanding of the two stage innovation adoption model, we can now turn to four broad dimensions and their variables that provide a basis for the organizational innovation for CASE to succeed. The four dimensions and their variables are listed below as they relate to overall CASE penetration (Rai and Howard, 1993). See Appendix B.

1. Structure of the ISD
    - Size
    - Functional Differentiation
    - Extent of Specialist Knowledge
    - Job/Role Design
2. Management Processes
    - Environmental Scanning
    - Training
    - Justification of Innovations (CASE)
3. Management Support Factors
    - Institutional Leadership
    - Champions of Technology (CASE)
4. Corporate Systems Delivery
    - Performance Gap of the ISD
    - Role Uncertainty of the ISD

1) Structure of the ISD

Size

There tends to be a direct relationship between the size of the organization and the inclination to adopt new innovations. Essentially, larger firms tend to absorb more innovation merely because of slack resources such as financial and human reserves. With a large reserve of capital, for instance, it makes it possible for an organization to pursue innovation. CASE technology is an excellent example of this. Because CASE tends to be costly and time consuming, only the larger organizations are able to commit the resources necessary to explore such an innovation. Comparatively, smaller ISD's are generally poorer in financial and human resources and can not devote what they do have to the possible risks associated with innovation. It is interesting to note however, that there is some evidence that contradicts this general pattern. Often small firms establish themselves by being innovators within their respective industries, rather than shying away from innovation. And larger firms, as they grow tend to become more conservative, resulting in viewing innovation as a threat to their already established technologies and not desiring to continue innovation. Because of this contradictory evidence, no direct link between the size of the organization and innovation can be definitively proven, however a compromise could be argued. Size can promote innovation up to a point, after which diminishing returns begin to set in (Rai and Howard, 1993).

Functional Differentiation

Functional differentiation encompasses the extent to which the ISD is divided into additional subunits. Essentially, the more differentiation means more special interest groups and more demands for additional technology (Rai and Howard, 1993). Each subunit will pursue its own interests and goals, and to do so will be required to demand more innovation from the entire ISD, thus making the ISD as a whole more innovative. For example, if one segment focuses on analysis while another focuses on coding, each will demand different innovations within CASE technology. The analysis team, for example, will require diagraming tools while the coding team will require code generation tools, and both will demand CASE technology in general. Essentially, functional differentiation perpetuates innovation, such as CASE.

Extent of Specialist Knowledge

The current state of knowledge residing within the organization and the degree to the wanting of additional knowledge and skills can provide inertia for innovation. If the organization has these aspects, they are more likely to acquire and experiment with sophisticated CASE tools.

Job/Role Design

Often innovation is seen as a threat to people within the organization. This is because innovation is often seen as eliminating present tasks and skills as well as disrupting work flows. Resistance, therefore, is often inevitable if employees view such innovation in this manner. It is imperative that the employees are receptive to new tools and techniques (Shafer and Shafer, 1993). To smooth the transition, rotating employees through different functions and/or redesigning their jobs around the innovation can help to alleviate the stressful situation and result in a more accepting attitude to the innovation and change. This is due to the fact that after job rotation or role redesign, one can better appreciate how everything relates to one another. Developing a larger skill base and crossing functional boundaries can help employees accept that the innovation is beneficial to the organization and therefore should be embraced. For example, a programmer that feels threatened by a diagraming CASE tool that automatically generates code, could be rotated to the analysis phase and be offered to work with such a tool. This would allow him or her to see the benefit to the organization and hopefully accept the innovation more willingly, assuming his or her job would not be completely replaced. By using job rotation and role redesign techniques, therefore, can assist in the adoption of innovation.

2) Management Processes

Environmental Scanning

The more communication and observation that an organization has with the external environment, the more aware they will be of emerging innovations that could benefit their operations (Rai and Howard, 1993). "In a period of declining production, price volatility, high interest, high overhead, and environmentalism, organizations continue to formulate strategies to survive and compete effectively" (Miranda and Tellerman 1993, p.33). On such strategy is environmental scanning. Without knowing one's external environment a business can not possibly hope to respond to its customers needs effectively. Scanning the environment for new innovations, such as CASE, is critical for success. When CASE tools first appeared on the market, those firms that began experimenting with them first gained enormous advantages over the latecomers. While some end up rejecting the technology later, they at least took the chance in hopes of positive gains. Not all risks result in a positive return, but only those who attempt them will ever reap the rewards.

Training

Training can accomplish two crucial items when addressing innovation. First, it imparts new and necessary skills upon the employees to deal with change. Second, it can assist in removing the fears associated with new technologies. It can overall impart positive feelings about the new changes. Training in the case of CASE is critical. The sheer magnitude of capabilities that CASE offers demands extensive training simply to operate the software. Even more importantly, however, is the training necessary in developing a formal methodology to follow. CASE in and of itself is not a methodology, it merely supports a methodology and prior standards of development. Research has shown that the establishment of a methodology to follow in systems design work is critical to the adoption of CASE (Rowe, 1993). Without it, CASE will never proceed beyond the experimentation stage. Training in these "soft" areas is often the factor that can make or break an exploration into CASE technology.

Justification of Innovations (CASE)

For innovation to be successful, the organization must justify the expenses as necessary and beneficial in the long run for the firm to succeed. Risk averse firms are not likely to devote adequate resources to experiment with innovations, such as CASE technology, because the payoff is too uncertain. While CASE tends to require a substantial investment of money and time, the potential of redesigning the entire traditional systems development process and substantially improve the performance of the system and posture of the business in its industry is great. However, relying on strict cost-benefit and ROI analysis often leads to the rejection of such innovation (Rai and Howard, 1993).

3) Management Support Factors

Institutional Leadership

Without clear and strong leadership, innovations may never make it out of the initiation stage, or may never be initiated at all. Proper leadership can guide innovation by creating an appropriate cultural base, meaning that leadership can direct how innovation will take place and evolve necessary for the organization. Additionally, strong leadership can help to overcome the political obstacles often associated with innovation. Because CASE is costly and raises fear among employees, clear and definitive leadership is critical for CASE innovation.

Even more important, management must view information systems as strategic to the business. "If the belief is that the IS function's contribution to the business is non-strategic, the management vision will provide solutions at the lowest cost available with a short-term view" (Miranda and Tellerman 1993, p.34). Strategic-based IS shops, on the other hand, have a longer-term vision. In this instance, management views innovation and new technology as a source of benefits to the organization. CASE is looked upon as a means to increasing IS value contribution to the organization (Miranda and Tellerman, 1993). Essentially, innovations such as CASE must be supported unequivocally by upper management and must view their development as strategic to the business, otherwise the innovation has no hope of succeeding.

Champions of Technology (CASE)

A champion of an innovation is one that vigorously promotes its use. While it may be the leaders just mentioned, it is often someone from lower ranks in the organization. It is their role to bring to the attention of upper management and convince them that the innovation would benefit and be feasible for the organization, or that it should be at least considered and explored (Rai and Howard, 1993). Additionally, champions of innovation can also help to overcome the resistance among the employees within the ISD. In fact, if the champion can convince his/her peers initially of the possible innovation, such as CASE, it may make it easier for that person to sell the idea to upper management, drawing on the support of his/her associates. Thus, enthusiastically championing CASE will greatly improve the odds that the technology will eventually permeate the entire organization.

4) Corporate Systems Delivery

Performance Gap of the ISD

If current methods and technologies do not allow the organization to meet its expected performance standards, new technologies may be explored as a solution to the shortfalls (Rai and Howard, 1993). The gap can result for many reasons, including rising output targets, declining performance, or additional threats from competitors. For example, if new system components are continually being completed late, research into CASE may result in hopes of decreasing development times. The appearance of performance gaps tends to be an excellent predictor of innovation.

Role Uncertainty of the ISD

Environmental instability leads to uncertainty of the ISD. This instability and uncertainty often perpetuates the innovation process because firms become more future oriented. The future orientation in turns forces innovation as a means to stay competitive and agile in the unstable environment. Therefore, ISD's that face a high degree of uncertainty should initiate the exploration of innovations such as CASE technologies.

The Principal Financial Group

As a real world application of these variables, we can look at

The Principal Financial Group, headquartered in Des Moines, Iowa. (Note: All information pertaining to The Principal Financial Group was obtained through an interview with Randy Roth, senior systems analyst, and through my work experience as an information systems intern during the summer of 1994.) The Principal is major worldwide insurance company. Regarding the size variable, The Principal is very large with a workforce of over 7,000 employees. In opposition to the theory, however, The Principal has at least explored CASE technology willingly, despite its large commitment and investment to past technologies. They have, however, opted against implementation primarily because it was felt the technology actually slowed development time and did not significantly improve the quality of the resulting system. Never-the-less, The Principal did not allow its size to prevent the experimentation with CASE technology.

In light of The Principal's non-use of CASE, some of the remaining variables cannot be analyzed simply because they have not employed, for example Job/Role Design. For this reason, only a few variables will be discussed. Environmental scanning, for example, is employed well at The Principal. This company continually monitors its external environment and looks for innovations to adopt. In the competitive industry of insurance, The Principal has found it necessary to stay at the leading edge of technology and therefore actively monitors its environment for subtle changes that can affect its market position.

The Principal also justifies its innovation as necessary and strategic to the firm. If they didn't, they never would have begun experimenting with CASE technology to begin with. No immediate ROI or payback periods are imposed, rather the idea of trial and error and accepting of losing money in hopes of future gains are supported.

Simply looking at the already mentioned variables relating to The Principal, it is obvious that there exists strong and clear institutional leadership. It has been upper management that has initiated much of the exploration of CASE, and has justified it as necessary for success. Without this support, CASE would never have been considered at The Principal.

Finally, because the external environment is instable, the role of The Principal's ISD tends to be uncertain. It is clear of course that their role is crucial and strategic to the organization, but exactly how to go about providing the premier support the organization needs is difficult to articulate. Because of this the ISD at The Principal is always willing to explore new innovations as a means for it to fulfill its role in the organization.

Conclusion - Organizational Context

During the CASE initiation phase, management needs to promote experimentation, without the threat of punishment for failures or risks, as well as no insistence on immediate paybacks. A structured methodology needs to be in place or developed in this phase as well. Establishing open lines of communication within the ISD as well as with the external environment and rotating and redesigning jobs/roles are also critical. Also helpful, though not as essential, are an influential and vocal champion of CASE and strong, clear institutional leadership.

During the implementation phase, traditional methods of development need to be phased out. While the underlying methodology may not change significantly, the old shortcuts and rule breaking need to end. A solid project management approach assigning clear responsibilities and demanding milestones is necessary to ensure proper implementation. Of upmost importance in this stage are extensive quantities of technical as well as methodology training programs. Interpersonal skills, cooperation, teamwork, and communication need to be emphasized as well. See Appendix C for a summary of CASE organizational context innovation variables.

B: Analyst's Skills Context

Organizational critical success factors, however, are only one perspective in the successful adoption of CASE technology. Another, equally important aspect, are the analyst's skills. In fact, a number of studies report that skills such as information gathering, project planning, and human relations are the most important skills to an effective systems analyst (Sumner and Ryan, 1994). The technical skills, such as programming, are usually considered less important. Additionally, analysts must posses a thorough understanding of the business and its requirements, an ability to communicate clearly, and a willingness to cooperate collectively in a team environment. All of these skills are not part of any CASE software package. Like organizational variables, they are requirements that need to be in place within an organization even before CASE technology can be explored.

These analyst skills (competencies) can be broken down into the general life cycle stages of analysis, design, detailed design, and implementation (Sumner and Ryan, 1994). Research conducted by Mary Sumner, Professor in the School of Business at Southern Illinois University and Terence Ryan, Assistant Professor at Indiana University an interesting look into these skills. Eighty-eight members of a CASE Users' Group in St. Louis were asked to list critical success factors in systems development, and then assess the importance and degree of difficulty of achieving each of these factors. The result of the study can be seen in Appendix D and will be explained next.

The highlighted skills in each of the tables reflect the key variables/skills necessary for successful systems development. Regarding analysis, both the ability to involve the client in the development process and the ability to set the boundary (scope) of a project were viewed as important and difficult to achieve. In terms of design, the ability to understand the client's business was perceived as both important and difficult to achieve. The ability to establish effective communications between the designer and user was viewed as important in detailed design, while the ability to coordinate project activities so that tasks are completed within time and cost constraints was viewed as difficult to achieve. Finally, examining implementation it was found that the ability to obtain customer acceptance of the final product as being important, while the ability to manage the process of organizational change was viewed as difficult to achieve (Sumner and Ryan, 1994).

The list of competencies displayed in the tables of Appendix D form the basis of the necessary analyst skills for successful systems design. These, coupled with the necessary organizational components previously discussed, now form a broader base of aspects necessary for not only successful systems design, but also the adoption of new technology such as CASE. To explore further, we can examine how well CASE can support these necessary analyst skills.

Having compiled the most important competencies as ranked by the eighty-eight member survey, they were then asked to assess the impact CASE tools had on achieving these. As you can see from table 5 in Appendix D, the CASE users did not view CASE technology as having a positive impact on achieving these critical success factors. In fact CASE tools made two competencies, the ability to involve the client in the development process and the ability to establish effective communications between the designer and the user, actually made it more difficult to achieve these factors(Sumner and Ryan, 1994).

Conclusion - Analyst's Skills Context

These somewhat disheartening results force analysts to wonder whether or not CASE is simply a technology that automates a series of processes rather than actually improving the underlying overall process. Regardless of the answer, it is evident that both key organizational variables and key analyst skills need to be present for CASE adoption and successful systems development. Whether or not CASE can actually improve the quality and efficiency of these developed systems is another question to be explored momentarily, but first let us finally turn to the technical aspects of CASE.

C: Technical Context

Technical issues include what CASE vendors can design into their products to make CASE more workable. One problem often associated with systems development is that user requirements for the system tend to be extremely volatile. After the requirements for the new system are laid out in the planning and/or analysis phase, they begin to expand in scope. It has been estimated that growth rate is 1 percent per month. Therefore, a three-year project would would have a one-third increase in scope by the time the system was completed. The challenge for CASE (technically) is to support these frequent modifications to the plans and specifications. In some regards, CASE has been fairly successful in this area, but there still exists significant room for improvement (Jones, 1992).

Another important problem that CASE needs to address is the high degree of errors inherent in the SDLC. Often 25-50% of the cost associated with a new system is the time spent removing defects and maintaining quality control (Jones, 1992). CASE, therefore, needs to support advanced forms of defect tracking, defect removal, inspections, testing and overall quality function deployment (QFD) and total quality management (TQM). Thus far CASE tools have failed miserably in this area, with only a few vendors offering modest quality control features (Jones, 1992).

Software projects also tend to generate enormous amounts of paperwork. While such paperwork may be necessary, particularly for documentation purposes, it is still not made easier to deal with in most CASE packages. For example, the concept of templates for standard document types is missing, as well as some type of on-line, integrated repository of documentation. While some vendors have started exploring this area, much work must still be done (Jones, 1992).

An additional shortcoming of CASE that needs to be built into the package is the communication and coordination of functions between the team building the system. Because software development is highly labor-intensive, CASE must support the effective and efficient communication between the developers.

Plans, status reports, specifications, modifications and changes, source code and test cases must be available as needed across heterogeneous, distributed organizations. In this area, some CASE vendors are offering full network support for these functions (Jones, 1992).

As the backlog of projects continues to grow, the need for reusable components become insatiable. CASE tools need to support reusability at many levels, including project plans, specifications, documentation as well as program code. To date, this too is lacking in many CASE tools, however, with the coupling of CASE technologies with object-oriented technologies, the future looks bright. It may be possible to achieve 50 percent reusability or more in the near future (Jones, 1992).

Finally, software does not tend to age gracefully. To combat this, CASE needs to include tools for restructuring, reverse engineering, and re-engineering program code. This is still not available on many CASE packages.

While CASE provides many useful components and tools for analyst to use, there are still many significant areas not addressed.

For CASE to truly become a revolution in information systems design, these "technical" aspects must first become incorporated within the technology.

Implementation Issues and Suggestions

Assuming for a moment, you are in the position of management and must coordinate the implementation of CASE technology in you organization. What things ought you know? To explore this it is helpful to understand the concept of marginal utility. To illustrate, let's use the utility derived from eating brownies. See Appendix E-1 and E-2.

As you can see, each additional brownie provides additional utility (satisfaction) up to a point, at which it begins to level off and finally decline. This appears around the eighth brownie. The picture becomes even clearer when you look at the marginal utility curve (Appendix E-2). Again, each brownie gives satisfaction, but at an ever-decreasing rate, until eventually the line crosses the x-axis (the eighth brownie), giving a severe stomach ache.

Armed with the concept of marginal utility we can now turn to

CASE. Once an organization finally begins using CASE, they do so in phases. The first actually does not deal with CASE at all, the first phase is simply using a structured methodology in systems design (Christoff, 1993). This means rigidly following the SDLC and the basic principles of design.

After an organization learns to exploit the power of a structured methodology, the next step is to attempt to use technology to assist in these efforts. This usually involves CASE, but only on a superficial level. The analysts learn that CASE provides an easy way to draw diagrams, document some work, and so on.

Finally the organization may fully utilize CASE (I-CASE) and use it in every phase of the life cycle from planning to support. So, how does marginal utility and particularly managements role fit into the whole implementation scheme of things? The fit appears as a problem, that is that organizations tend to get stuck in one of these stages and cannot move forward. It is management's responsibility to recognize this, and facilitate the progress.

Applying utility to this problem (see Appendix E-3) we can see that the benefits (satisfaction/utility) from each phase continues to rise. So why is it that organizations cannot seem to move ahead of their current phase (often the first or second)? The answer in the marginal utility curve (see Appendix E-4). As you can see, the marginal utility for CASE tools drops below the x-axis in each phase, similar to that of the brownies (Christoff, 1993). The point at which the curve drops below the x-axis is also the peak of total satisfaction. Additional rigor in attempting to move to the next phase only seems to make matters worse. No wonder organizations never progress, they have no incentive to do so, just as no one will eat a ninth brownie because there is no incentive to do so.

However, one must not give up so soon. As the curve suggests, if an organization persists eventually the utility becomes positive again and the total utility again rises. This is like saying that if you persist to the eleventh brownie, they will start to taste good again. One aspect to point out however is the difference in curves between brownies and CASE. With brownies the marginal utility curve slopes down. This is true with most commodities (Christoff, 1993). The first and additional units give less and less satisfaction. The marginal utility for CASE, however, is different. It has a positive slope first and then develops the traditional negative slope. The significance of this is that initially analysts are seduced by the technology, only to be greatly disappointed later. For example, the ease at which the developer can construct diagrams is appealing at first. He or she is motivated not by management but by their own intrinsic motivation to continue.

At this point, management intervention would be counterproductive. However, once the peak is reached, additional utility, while still growing totally, begins to slow down. Management's role at this point is to point out that while satisfaction is slowing down, it is still growing. The analysts need to continue to be motivated and management needs to take on a cheerleader role. The overall benefits, rather than each successive step, need to be emphasized (Christoff, 1993).

At the point where the utility crosses the x-axis, however, becomes extremely critical. Now each additional level or rigor has negative benefits, the analyst could do the process the old way and do it better. At this stage management must dictate continued use to assure passage on to the next stage. As the curve suggests, things will get better. A type of "utility-faith" must be accepted, and it is managements role to provide this.

A Case Study of CASE

The Principal began exploration of CASE tools in late 1989, with the hope of improving its systems development process through faster development times and higher quality systems. Various CASE products were selected for testing by the Research and Development Group within their IS department. Included for analysis were Texas Instruments Information Engineering Facility (IEF), Application Development Workbench (ADW), and Intersolv's Excelerator.

Initial analysis of new technology begins with an examination and testing of functions of new tools, followed by the necessary requirements the technology is to fulfill. A brief white paper to top management explaining the possible benefits of such technology is written, and if accepted the project moves forward. This whole preliminary process is often very short, as was the case with CASE tools.

A small pilot project was initiated using ADW, and utilized the entire life cycle features of ADW, from analysis to code generation. The results, according to Senior Analyst Randy Roth, were "relatively successful". Successful in that a deliverable system that met the end users requirements was produced. No significant problems were encountered, though no particular benefit could be noted related directly to CASE tool use compared to a traditional approach.

Five more pilot projects were initiated, again utilizing the full life cycle. The results of these were similar to the first pilot project. An acceptable system was generated, but no significant benefits were articulated as coming from CASE. In fact an upgrade in software versions even began to drain some productivity because of the increasing complexity of additional functions and a higher learning curve. The end result was that after only a few years limited use of CASE technology, The Principal dropped CASE altogether except for limited use of upper-CASE tools used for data modeling.

According to Roth, "CASE tools attempt to speed up an existing process. This, however, is a faulty assumption. It would be expected that the first phases (analysis and design) would actually require more time because of the importance of these phases, while only the later phases (coding) would be sped up. If you spend adequate time in analysis and design, coding should be about ten percent of the overall work. And if you're working to speed up only ten percent of your work, you're not fixing the real problem. The real problem is the method you use to get there. I don't believe at this time that CASE technology is a valid approach to accomplish systems development."

The experiences at The Principal have not been isolated. Numerous companies have explored CASE tools and have ended up abandoning them. While CASE has proven to produce functional systems, no significant additional value has been observed. And in many cases, only additional headaches have been observed.

Conclusion

So where does this leave an organization considering CASE tools, or an organization experiencing problems with CASE currently? It is helpful to examine the overall picture with a socio-technical systems approach. This approach views organizations as being made up of both technical and social aspects (Sumner and Ryan, 1994). The work system must optimize both the technical and social aspects in order for the overall system to contribute to achieving the organizational goals.

The technology of a work system includes the tools, methods and physical conditions for work. The social aspects consist of roles played by people and the interactions between these roles. Both the technical and social aspects of a system interact dynamically (Sumner and Ryan, 1994). The "big picture" question is whether or not CASE supports the aspects of the socio-technical perspective.

Through this analysis, it is evident that CASE supports many of the technical aspects of systems design, some better than others, though it does still offer some support. For example, most CASE tools can incorporate process and data modeling techniques that are a part of structured analysis and design methodologies, as well as support code generators and other technical components. Social aspects, however, of systems design, are not well supported by existing CASE tools. Role definitions, communication networks, and personal preferences are severely lacking in CASE technology.

Given the goal of information systems development is to design and improve information systems, and to do so requires both technical and social aspects so be supported, CASE tools may be of limited value. While significant strides have been made to improve the quality and timeliness of information systems development through the use of CASE tools, the technology remains in its infancy.

Works Cited

Christoff, Kurt A. 1993. "CASE, Marginal Utility . . . and Brownie." Information Systems Management pp. 66-69.

Jones, Capers. 1992. "CASE's Missing Elements." IEEE Spectrum pp.38-41.

Miranda, Max H. and Tellerman, Naomi A. 1993. "Corporate Downsizing and New Technology: The Impact on IS and CASE Tool Use." Information Systems Management pp. 32-38.

Rai, Arun and Howard, Geoffry S. 1993. "An Organizational Context for CASE Innovation." Information Resources Management Journal pp. 21-34.

Rowe, Joyce M. 1993. "Can Enforced Standardization Affect Case Usage?" Journal of Systems Managment pp. 29-33.

Shafer, Linda I. and Shafer, Donald F. "Establishing a CASE Toolbox: 15 Steps to Selecting CASE Tools." Information Systems Management pp. 15-19.

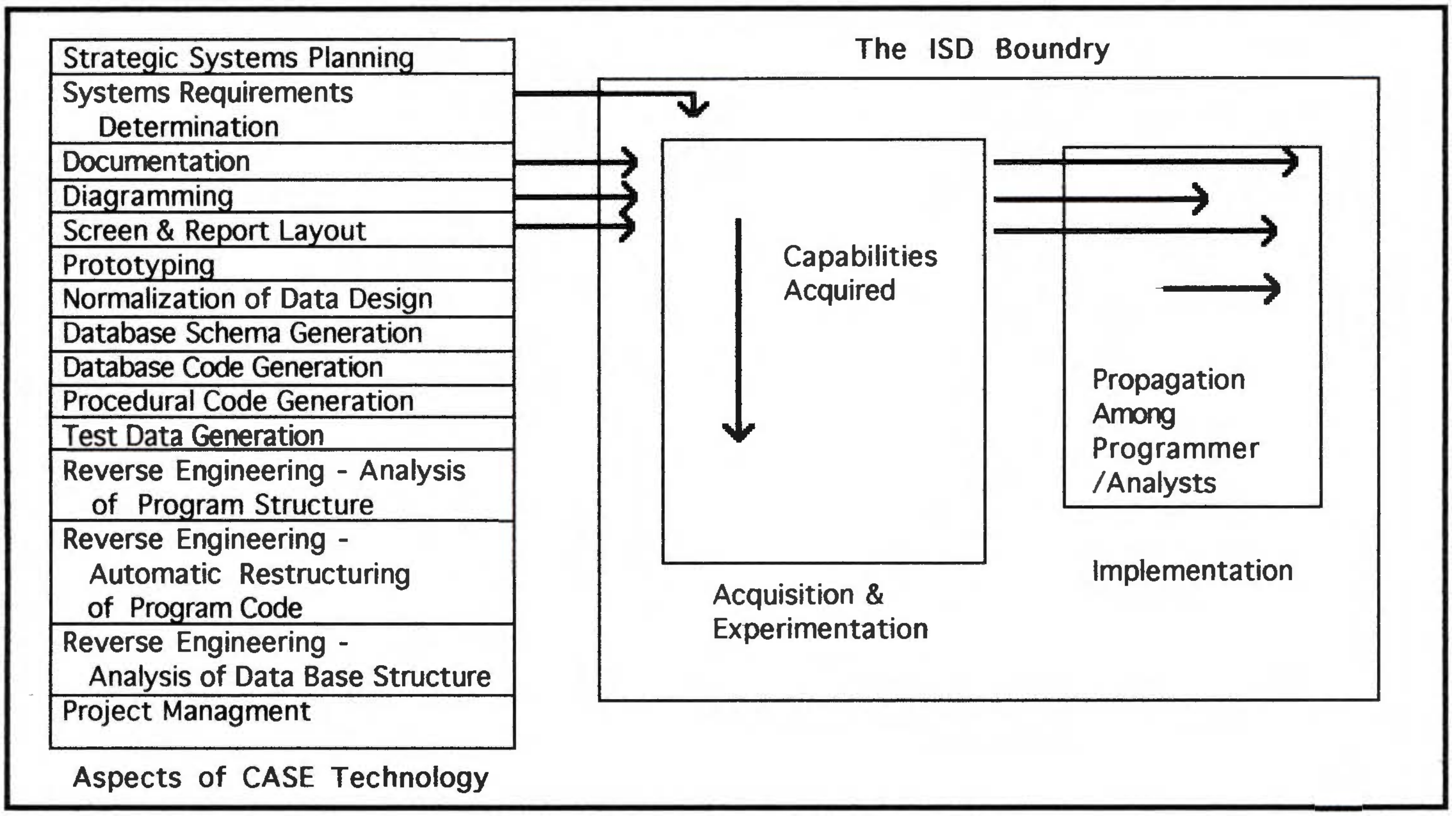
Steinber, Geoffrey and Baram, Giora 1992. "An Investigation of CASE Software Integration." Journal of Systems Management pp. 20-22.

Sumner, Mary and Ryan, Terence. 1994 "The Impact of CASE: Can it Achieve Critical Success Factors?" Journal of Systems Managment pp. 16-21.

Whitten, Jeffrey L. and Bentley, Lonnie D. and Barlow, Victor M. Systems Analysis and Design Methods Burr Ridge: Irwin, 1994.

Yellen, Richard E. 1992. "What Do Users Really Think About CASE?" Journal of Systems Management pp. 16, 39.

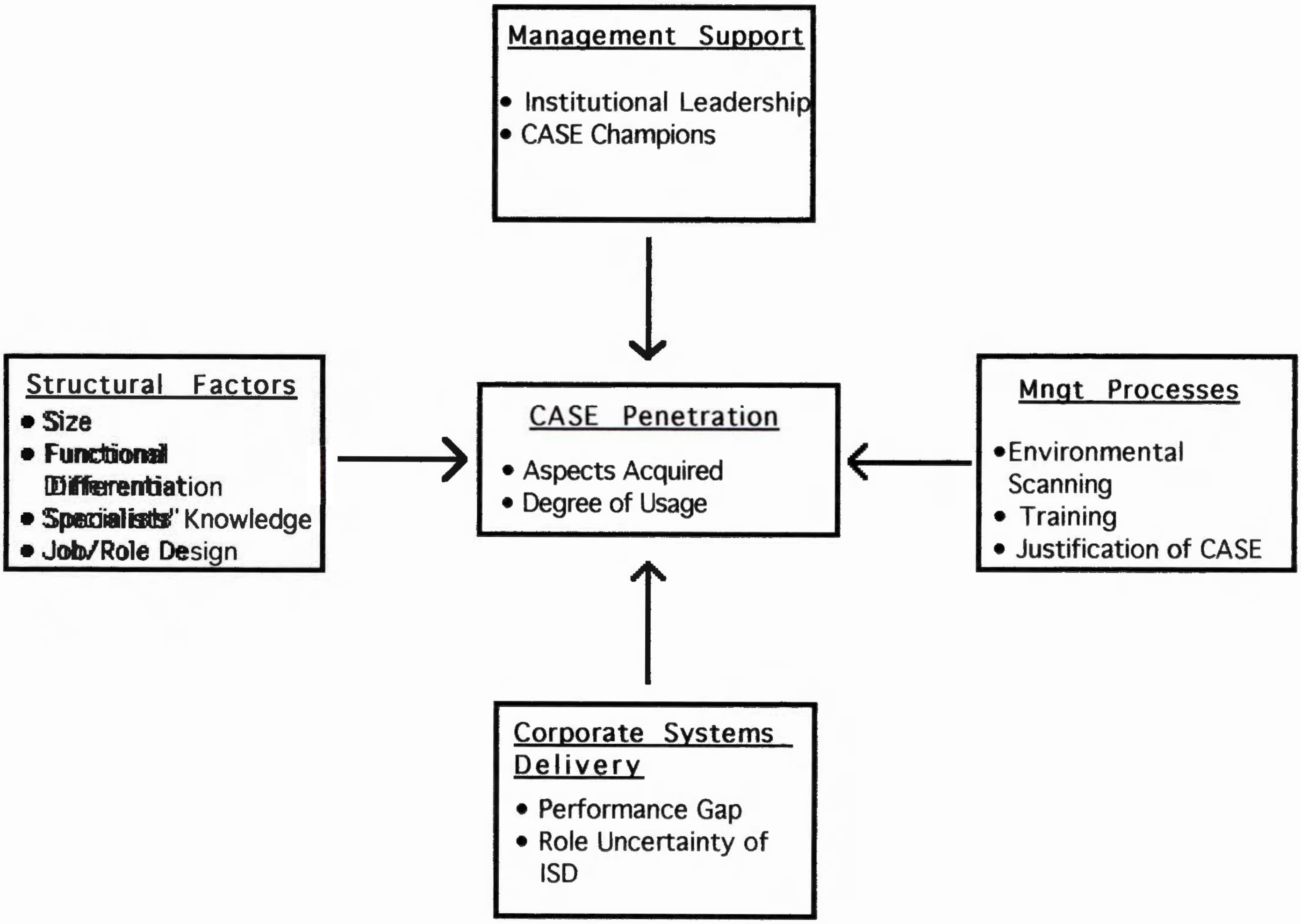
Appendix A



A Stage Model for CASE Adoption by ISDs

(Rai and Howard, 1993)

Appendix B



Correlates of CASE Penetration

(Rai and Howard, 1993)

Appendix C

Influence	Direction	Explanation
Size of ISD	+-	Size usually implies slack resources for experimentation. After a point, though, excessive size causes rigidity because of difficulty in changing complex entrenched work procedures.
Specialist Knowledge	+	Specialization allows programmer/analysts to learn up to and through state-of-the-art, at which point they become aware of innovative practices that are at the fringe of knowledge.
Functional Differentiation	+	Differentiated work groups of programmer/analysts lead to specialization, with innovative outcomes as explained above.
Job/Role Rotation	+	Ongoing job role rotation promotes flexibility, prevents entrenchment, thus lubricating attitudes in preparation for a change to CASE work methods.
Methods and CASE Training	+	Knowledge of structured methodologies promotes CASE use, as most CASE tools are designed to serve as "methodology companions."
External Information Sources	+	Many external communication channels enhance awareness of the newest technologies and CASE tools.
Justification	-	Innovation is risky, and will not prosper in a risk-averse organizational culture.
Institutional Leadership	+	Clear top management vision of the role of the ISD in the overall organization leads to clear appreciation of the value of systems innovations to the entire business.
CASE Champion	+	Vocal and powerful believers in CASE create an open environment for CASE experimentation and aid in resource-acquistion to pay for implementation.
Performance GAP of ISD	+	Managers look for innovations that can improve productivity as a fix for an ISD's performance shortfall.
Role Uncertainty	-	Hard to sell investment in uncertain, long term payback innovations like CASE when the future of the ISD is in question.

Summary of a Positive Organizational Context for CASE Innovation
(Rai and Howard, 1993)

Appendix D

Table 1: Competencies in Requirements Analysis

Competency	Importance	Degree of Difficulty
Ability to involve the client in the development process	4.77	4.15
Ability to obtain support for the project	4.85	3.54
Ability to set the boundary (scope) of a project	4.38	4.42
Ability to identify the problem/opportunity within the boundary of a project	4.23	3.54
Ability to decide whether it will be worthwhile to pursue solution of the problem/opportunity	3.92	3.12
Ability to choose the team who will do investigation and modeling	3.23	3.00

(Sumner and Ryan, 1994)

Appendix D
(cont'd)

Table 2: Competencies in Systems Design

Competency	Importance	Degree of Difficulty
Ability to understand the client's business	4.54	4.35
Ability to communicate the results of investigation & modeling activities to those who approve them	4.46	3.96
Ability to investigate the existing system, its environment, and its functions	3.46	3.23
Ability to create alternate "good" logical models to represent possible solutions to problem/opportunity	3.77	3.73
Ability to produce a "good" logical model (i.e. consistent, complete, valid, flexible) of the existing system	3.23	3.62

(Sumner and Ryan, 1994)

Appendix D
(cont'd)

Table 3: Competencies in Detailed System Design

Competency	Importance	Degree of Difficulty
Ability to establish effective communications between the designer and user	4.61	3.92
Ability to coordinate project activities so that tasks are completed within time and cost constraints	3.92	5.04
Ability to document system design specifications accurately and completely	3.92	3.42
Ability to create modular, flexible program design specifications	3.77	3.65
Ability to construct a simple, effective user interface in the design of reports and screens	3.69	3.19
Ability to prototype the design of reports and screens so that user requirements are defined	3.38	3.00

(Sumner and Ryan, 1994)

Appendix D
(Cont'd)

Table 4: Competencies in Systems Implementation

Competency	Importance	Degree of Difficulty
Ability to obtain customer acceptance of the final product	4.85	3.38
Ability to maintain effective communications between the analyst and user	4.46	3.69
Ability to develop and implement an effective training program	4.31	3.27
Ability to design and implement effective testing strategies	4.15	4.04
Ability to manage the process of organizational change	3.92	4.65
Ability to develop thorough systems design documentation	3.04	2.73

(Sumner and Ryan, 1994)

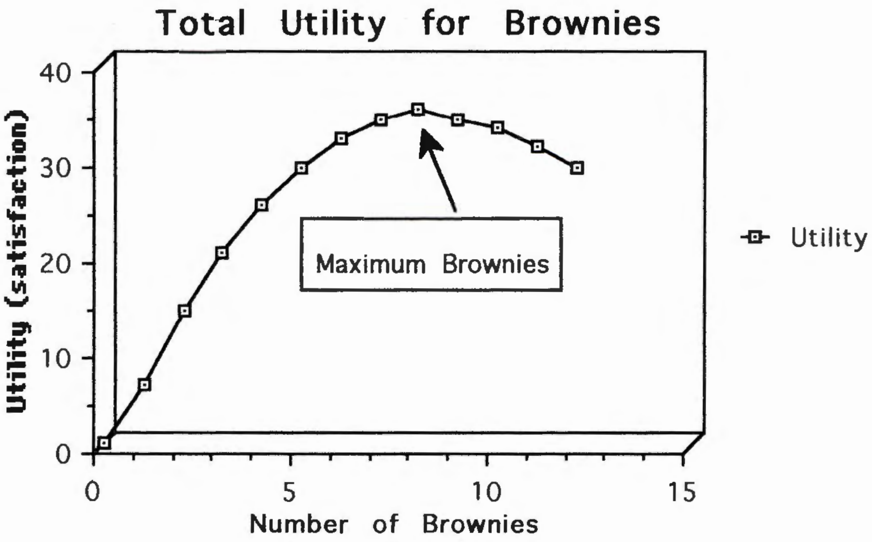
Appendix D
(Cont'd)

Table 5: Impact of CASE on "Most Critical" Factors

Competency	Raw Score	Mean Impact Score
Ability to obtain support for the project	4.76	0.0
Ability to understand the client's business	4.53	0.0
Ability to obtain customer acceptance of the final product	4.47	0.0
Ability to involve the client in the development process	4.47	-1.0
Ability to maintain effective communications between the analyst and user	4.41	0.0
Ability to set the boundary (scope) of a project	4.35	0.0
Ability to establish effective communications between the designer and the user	4.29	-1.0
Ability to coordinate project activities so that tasks are completed within time and cost constraints	4.18	0.0

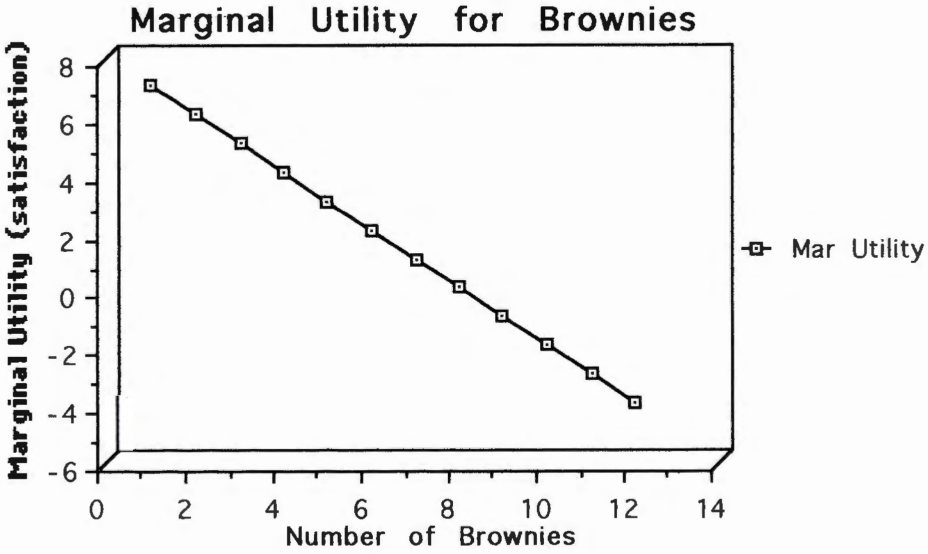
(Sumner and Ryan, 1994)

Appendix E-1



(Christoff, 1993)

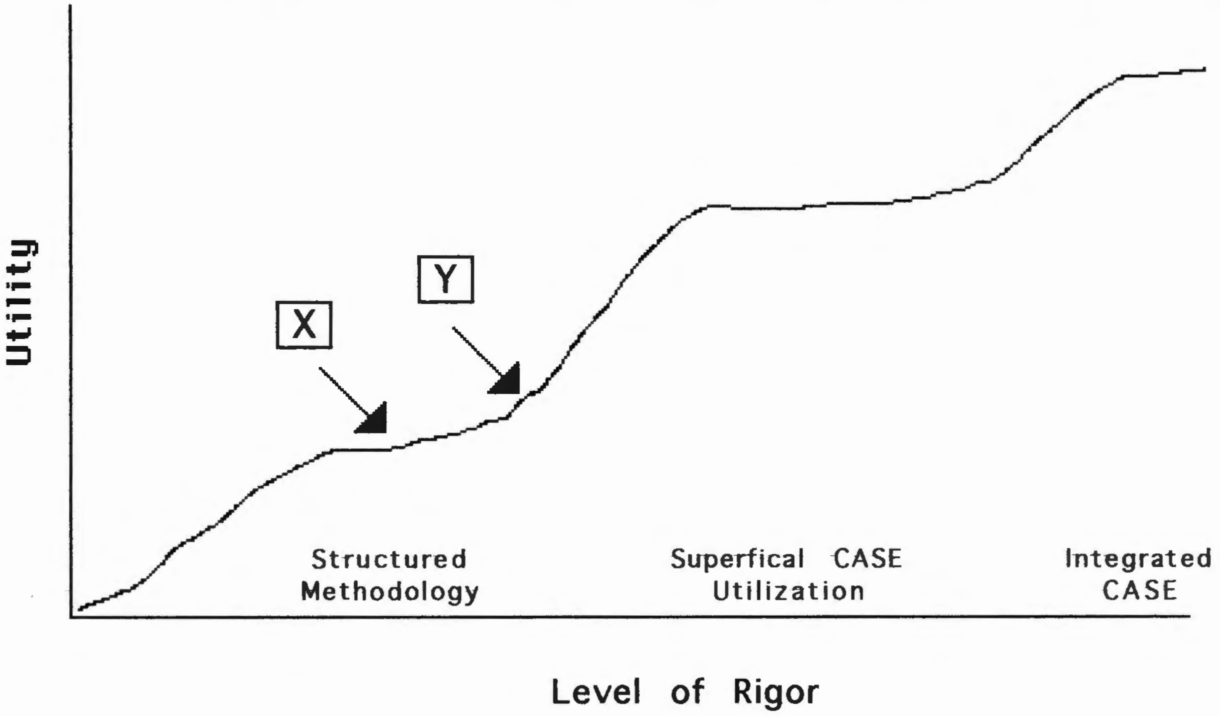
Appendix E-2



(Christoff, 1993)

Appendix E-3

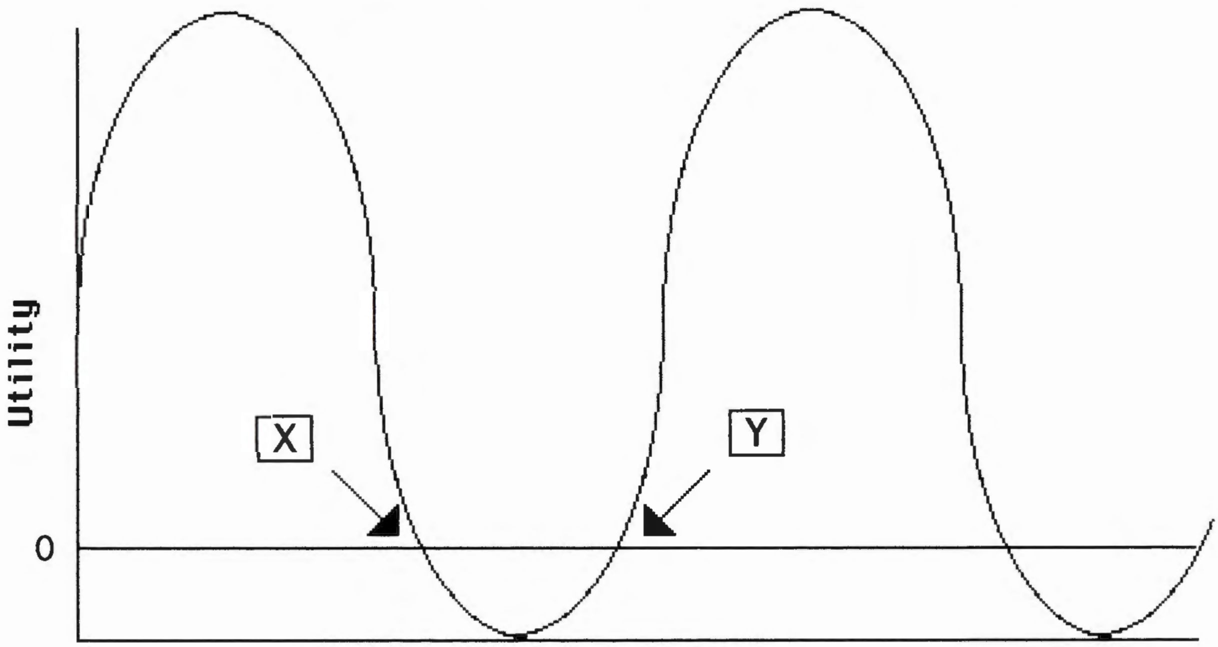
Total Utility for CASE Tools



(Christoff, 1993)

Appendix E-4

Marginal Utility for CASE Tools



Level of Rigor

(Christoff, 1993)