An Organizational Context for CASE Innovation

ARUN RAI
Southern Illinois University

GEOFFRY S. HOWARD
Kent State University

Practitioners and researchers have suggested that Computer-Aided Software Engineering (CASE) technology can significantly improve productivity of programmer/analysts and quality of application systems. Before CASE can succeed, though, a particular organizational context must be created within an Information System Department. The paper describes the necessary context. This paper first briefly discusses the present state of CASE technology. A model for CASE adoption by Information System Departments is developed by drawing upon past research in organizational innovation, IS implementation, and systems development. An interview-based methodology is used to determine the most favorable context for implementing CASE technology and to perform a preliminary test of the proposed model. Implications for practice and avenues for future research are identified.

CASE Technology: An Innovation in Systems Development

Many Information Systems Departments (ISDs) are carefully reevaluating their systems delivery strategies. The potential of information systems to enhance the competitiveness of organizations has been echoed in the literature. Users are looking to their ISDs to create products that can be delivered (and modified) quickly to meet the “requirements volatility” of a competitive business environment (Cooprider and Henderson, 1991). Increasingly line and corporate management are recognizing that integrated computer-based information systems can reengineer traditional business processes (Hammer, 1992). The infrastructure, processes, technologies, and job-skills required to build such systems are different from those needed for independent stand-alone systems built for individual functional areas (Rockart and Hofman, 1992).

The above challenge is compounded by performance pitfalls reported by many ISDs. Currently maintenance activities consume about 70% of the total time spent on applications. Large maintenance times substantially increase user-backlogs with the average backlog now estimated to be 30 months. It is also estimated that there is an equal, if not greater, invisible user backlog. This consists of applications that never get formally requested because of long lead times. If the present trend continues, ISDs may eventually find all development resources being consumed by maintenance related activities (Bachman, 1989).

CASE technology has been receiving increasing attention as it has been recognized as a possible means to enhance the productivity and performance by redefining the software development process. It has been documented as a technology that can increase competitiveness of organizations, bring about productivity gains, and reduce costs and lead times involved in systems development work (Feuche, 1989; Mar-
The broadest definition of CASE has been provided by Stamps (1987) as the “automation of anything a human does to software”.

Some tools support simple stand-alone functions such as documentation or diagramming while others adopt a development methodology and integrate several system development activities. Most methodologies today are based on the central premise of prototyping. Under this approach the system development process should proceed much more rapidly and iterate among the stages of analysis, design, coding and testing. The adaptive refinement of the system emphasizes user involvement, utility, and satisfaction (Alavi, 1984), reduces the time to build systems (Blum, 1986), and reduces the number of design defects (Jones, 1986).

“Front-end” CASE tools support logical design activities by providing graphic environments that support process and data modeling. Typical features offered by such tools include data flow and entity-relationship diagrams, data dictionaries, report and screens design, documentation generation etc. Excelerator, a PC-based product developed by Index Technology Inc., is a good example of a commercial front-end CASE tool (Whitten & Bentley, 1987).

It is important to effectively link front-end logical modeling with the back-end tasks of physical design and implementation. CASE tools with forward engineering capabilities start with logical data models and generate database code/schemas. Some forward engineering products facilitate procedural code generation in languages such as COBOL. On the contrary, reverse engineering products start with program code or database schemas and reconstruct logical models. These tools can be used for functions such as analysis of program and database structure, and automatic restructuring of program code. The DBMS CASE tools developed by Bachman Inc. exemplify these capabilities (Bachman, 1988).

Products that span all development functions—logical and physical design, testing (CAST -- Computer Aided Software Testing) and implementation—for individual application systems are referred to as full life cycle products. Increasingly ISDs are sensitive to the fact that they need to integrate their application systems portfolio. Computer-Aided Systems Engineering or Information Engineering products link the process of strategic systems planning with the development of the application systems portfolio (Inmon, 1988). These products provide a platform for the production of software and coordination among project team members. They also assist in constructing a data architecture—a key challenge in moving toward an integrated application systems environment. Examples of commercially available information engineering tools include Navigator by Ernst and Young, IEW by Texas Instruments Inc.

Emerging concepts such as object-oriented methodologies, reusable code, and software libraries are being gradually integrated into the range of capabilities of CASE. All of the examples discussed represent instances of support technology that can be employed to change the software development process, and consequently the nature of the products that are produced. Thus, CASE technology is a possible solution to achieve productivity gains, enhance systems quality, and build integrated systems that can reengineer traditional organizational processes.

In the context of the IS function, implementing CASE is a prime example of “business process redesign”. The technology can significantly change the way the fundamental responsibility of the ISD -- systems development -- is accomplished (Rockart and Hofman, 1992). It is important, then, to understand how to successfully initiate and implement CASE technology. As a first step, then, in developing an understanding of how to manage CASE, we set about constructing an explanatory model from pre-existing literature.

**Model Development**

Since CASE is an innovation in systems development, we draw upon past research in organizational innovation, IS implementation, and systems development to define an organizational context for an ISD that seems fertile for CASE success. In order to more fully understand and describe this favorable organizational context, we propose here a model for CASE penetration in ISDs. The model is a priori, deduced from existing literature. The veracity of this model and of the set of characteristics of an organizational context that would be favorable to CASE innovation were then investigated by conducting in-depth interviews with senior IS managers in numerous companies. The purpose of the interviews was to check the real-world validity of our literature-based model, and to verify that our proposed set of organizational context characteristics was complete. The interviews allowed us to conduct a detailed exploration of the enabling and inhibiting issues in moving toward a favorable CASE context.

Management of innovations is a primary concern of CEOs (Van de Ven, 1986). The increasing turbulence and competitiveness of organizations’ environments have made the identification, evaluation, and adoption of technological innovations a critical determinant of organizational performance (Zaltman, Duncan & Holbeck, 1973). Kwon and Zmud (1987) emphasize that the vast base of empirical studies on organizational innovation provides the appropriate reference discipline to conduct implementation research in an IS context. In fact, IS researchers are increasingly studying IS implementation from the context of organizational introduction of a technological innovation (McFarlan and McKinney, 1982; Zmud, 1984; Nilakanta and Scammel, 1990).

**Stages of CASE Adoption**

Organizational innovation is often viewed as a two stage process consisting of initiation and implementation (Thompson, 1969; Zaltman et al. 1973; Pierce and Delbecq, 1977; Rogers 1983; Van de Ven, 1986). Initiation is brought