

**RADICAL AND INCREMENTAL
INNOVATIONS IN SYSTEMS
DEVELOPMENT: AN EMPIRICAL
INVESTIGATION OF CASE TOOLS**

Wanda J. Orlikowski

April 1991

**CISR WP No. 221
Sloan WP No. 3283**

©1991 W.J. Orlikowski

**Center for Information Systems Research
Sloan School of Management
Massachusetts Institute of Technology**

Radical and Incremental Innovations in Systems Development: An Empirical Investigation of CASE Tools

ABSTRACT

This paper proposes a theoretical model for systematically accounting for differences in organizational outcomes occasioned by the adoption of CASE tools. Conceptually, the model draws on innovation research and research into the introduction and use of information technology in organizations. While drawing on prior theory, the model is grounded in and elaborated by empirical data, generated by two field studies conducted into the implementation of CASE tools in systems development. The paper argues that in order to understand the implications of CASE tools, it is necessary to distinguish between changes to the process of systems development and changes to its product. Results of the field studies show that the consequences of CASE tools are influenced by the motivation behind the decision to adopt the CASE tools, and the established structure and practice of systems development in place prior to the deployment of CASE tools. This understanding allows us to better explain and predict the influence of CASE tools, suggesting where the influence of CASE tools innovation is likely to be, and further, what challenges the innovation is likely to represent to the organizational and information system status quo.

Introduction: CASE Tools and Organizational Change

CASE (Computer-Aided Software Engineering) tools have stirred great interest in the research and practitioner communities as potential means for easing the software development and maintenance burden threatening to overwhelm information systems departments. The actual impact of these tools on development and maintenance activity, cost, and duration in the long term, are as yet unclear. Nevertheless, it is evident from a few studies and commentaries [Orlikowski 1989; Norman and Nunamaker 1989] that these tools have engendered a number of changes in the culture, structure, and practices of information systems departments. The extent and nature of these changes, however, appear to vary across organizations and over time. As yet, there are no good explanations for these changes and their variations.

In this paper, I propose a theoretical model for systematically accounting for differences in organizational outcomes occasioned by the adoption of CASE tools. Conceptually, the model draws on two different streams of research: (i) research on innovation--in particular, that dealing with different types of innovations [Dewar and Dutton 1986; Gersick 1991; Tushman and Romanelli 1985; Utterback and Abernathy 1975; Zmud 1984], and (ii) research into the introduction and use of information technology in organizations [Bjørn-Andersen, Eason and Robey 1985; Iacono and Kling 1988; Markus 1983; Orlikowski 1991a; Rousseau 1989]. While drawing on prior theory, the model is grounded in and elaborated by empirical data. The paper argues that in order to understand the consequences of using CASE tools, it is necessary to distinguish between changes to the process of systems development and changes to its product. Results of the field studies show that the consequences of CASE tools are influenced largely by the motivation behind the decision to adopt the technology, and the established structure and practice of systems development in place prior to the adoption of the technology.

In the next section I outline the theoretical framework with which I studied the implementation of CASE tools in organizations. This is followed by a discussion of the research methodology and a description of the two organizations where I conducted my field studies. I then discuss the research results that emerged from the data analysis, and show how these were informed by the theoretical framework, and how they led to the development of a more elaborate theoretical model. I conclude by discussing the implications of the model and results for future research into the relationship between CASE tools and information systems units, as well as for practitioners contemplating implementing CASE tools in their organizations.

Initial Research Framework

In this paper, I propose that CASE tools be seen as innovations in systems development, and that insight into their use and impacts can be gained by employing some of the concepts from the innovation literature. Research on organizational innovation represents a well-developed stream of concepts, theories, and results [Abernathy and Clark 1985; Henderson and Clark 1990; Leonard-Barton 1988; Myers and Goes 1988; Rice and Rogers 1980; Rogers 1983; Utterback and Abernathy 1975; Zmud 1984], and some have specifically addressed the case of information technology [Bayer and Melone 1989; Cooper and Zmud 1990; Brancheau and Wetherbe 1990; Leonard-Barton and Deschamps 1986; Moore and Benbasat 1989; Pennings 1988]. These various studies span a wide range of theoretical topics and research insights, and I will not dwell on them here. What I do want to draw from this literature are two important conceptual distinctions, one between process and product innovations, and the other between radical and incremental innovations.

Innovations in Process and Product

The work of every organizational unit can be broadly decomposed into three components: input, transformation, and output. The *product* is the output of the work: the objects or services produced through the transformation activities and delivered or sold to individuals or organizations beyond the unit. The *process* of the work is the means through which the input is transformed into the output: the activities and resources (financial, intellectual, material, social, and structural) needed to accomplish the work. The distinction between process and product innovations is a long-standing one in the innovation literature [Kazanjian and Drazin 1986; Utterback and Abernathy 1975; Zmud 1984], and it provides a useful starting point for thinking about systems development and the role of CASE tools within it. The work of systems development can be understood in terms of a set of inputs, transformations, and outputs--whether these are performed by an internal information systems department or an external software vendor. The *product of systems development* is the information system delivered or sold to clients (internal or external) typically comprising a package of software, data, manual procedures, documentation, and sometimes including hardware components, training courses, installation service, and support activities. The *process of systems development* includes the methodologies, practices, skills, standards, techniques, and tools utilized by systems developers to construct the information systems product.

Nature of Innovation

The innovation literature further recognizes that there are differences in the nature of an innovation implemented. Organizational units attempting to improve their efficiency or effectiveness may

attempt to innovate either their process or their product, or both. The type of innovation they adopt may be either *incremental* or *radical* [Dewar and Dutton 1986; Ettlie, Bridges and O'Keefe 1984; Gersick 1991; Pennings 1988; Tushman and Romanelli 1985]. The former implies a linear, cumulative change in a process or product, representing "minor improvements or simple adjustments in current technology" [Dewar and Dutton 1986:1423], while the latter are nonlinear, paradigmatic changes, representing significant departures from existing practice or knowledge. The categories of radical and incremental are intended as ends of a continuum representing the level of new knowledge embedded in an innovation, and not as exclusive categories. However, as Dewar and Dutton [1986:1423] note: "the middle values of this continuum are difficult to interpret." Some commentators propose finer-grained distinctions in innovation types [Henderson and Clark 1990; Meyerson and Martin 1987], but for the purposes of this paper, a dichotomous distinction suffices. This may need to be elaborated over time with more empirical work.

Dewar and Dutton [1986:1423] note that: "The distinction between radical and incremental innovations is easier to intuit than to define or measure." Because these distinctions rely on perceptions of new knowledge embodied in the innovation, individuals may differ in their classifications of innovations based on their experience, position, and expertise. While I draw on people's experiences with and perceptions of the innovations, I also examine the organizational infrastructure for clues as to the nature of a particular innovation. An innovation that is compatible with dominant assumptions and established organization practices and structures is likely to be incremental. Organizations that innovate incrementally refine or exploit the potential of their established processes or products, while their basic social relationships, rules, perspectives, and activity patterns remain the same [Gersick 1991:16]. An innovation which undermines dominant assumptions, departs from existing practices, and is incongruent with established structures, is likely to be a radical innovation.

An organization or organizational unit may attempt radical and incremental innovation in both its process and product. For example, consider the *process* of producing printed documents. From the perspective of a typist and his work unit, an electric typewriter would be an incremental innovation over a manual typewriter because it does not significantly alter the activities and resources needed to execute work. On the other hand, a word processing computer is likely to be a radical innovation over the manual typewriter because it requires different activities to do the work, necessitating dissimilar competencies, resources, standards, and procedures. With respect to *product*, consider computer software. New versions of a particular software product constitute incremental innovations for they enhance or extend existing features and capabilities. However, when a completely different software design is implemented it represents a radical innovation over the

existing one. For example, IBM's relational database product DB2 represented a radical innovation over its existing hierarchical one, IMS. Different knowledge, competencies, standards, and procedures are needed to work with it.

Towards a Theoretical Model

If we put these two dimensions together, we have a way of classifying the innovations that organizations may adopt, as well as some insight into their likely consequences. Figure 1 shows the two dimensions--locus of innovation and nature of innovation--with their respective values. When examining a particular innovation, this framework leads us to determine the influence a particular innovation has on a unit's process and product, and the amount of change in each implicated by the innovation. As I noted above, the distinction between radical and incremental innovation is often not clear-cut, and is particularly difficult to establish in the context of complex change and multiple groups. Where many players are affected by an innovation each may have his/her own interpretation of and response to the change--differences that are due to the diverse interests, positions, objectives, and knowledge of the players. Because perceptions of innovations may vary, it is important to determine (through interview, questionnaire, or observation) what changes the innovation has brought to key players' skills, roles, and activities. In addition, it is also necessary to calibrate perceived changes with actual changes in the institutionalized properties of the organization, that is, to establish what structural, procedural, and cultural changes have been occasioned by the innovation.

Implementing radical innovation is much harder than incremental innovation because it requires departures from existing norms, assumptions, knowledge, structures, and practices. Because incremental innovation does not require such departures, it is easier to accommodate and adapt to. Radical change is hard, not only because of the greater complexity, unfamiliarity, and uncertainty of the radical innovation, but also because most organizations are predisposed to stability and the status quo [Starbuck 1983]. Established structures, routines, and competencies create barriers to revolutionary change, and will require extensive and sustained effort to be transformed.

CASE Tools as Innovations in Systems Development

This more general model of innovation in organizations can be applied to the instance of CASE tools. CASE tools are targeted at the work of systems development and provide facilities such as design diagramming aids, code generators, repositories, debuggers, etc. The aim--judging from their designers' claims--is to reduce development time, improve productivity, increase consistency, facilitate reuse of prior work, and remove some of the more tedious development chores. These facilities most obviously and directly effect the execution of systems development tasks. As an

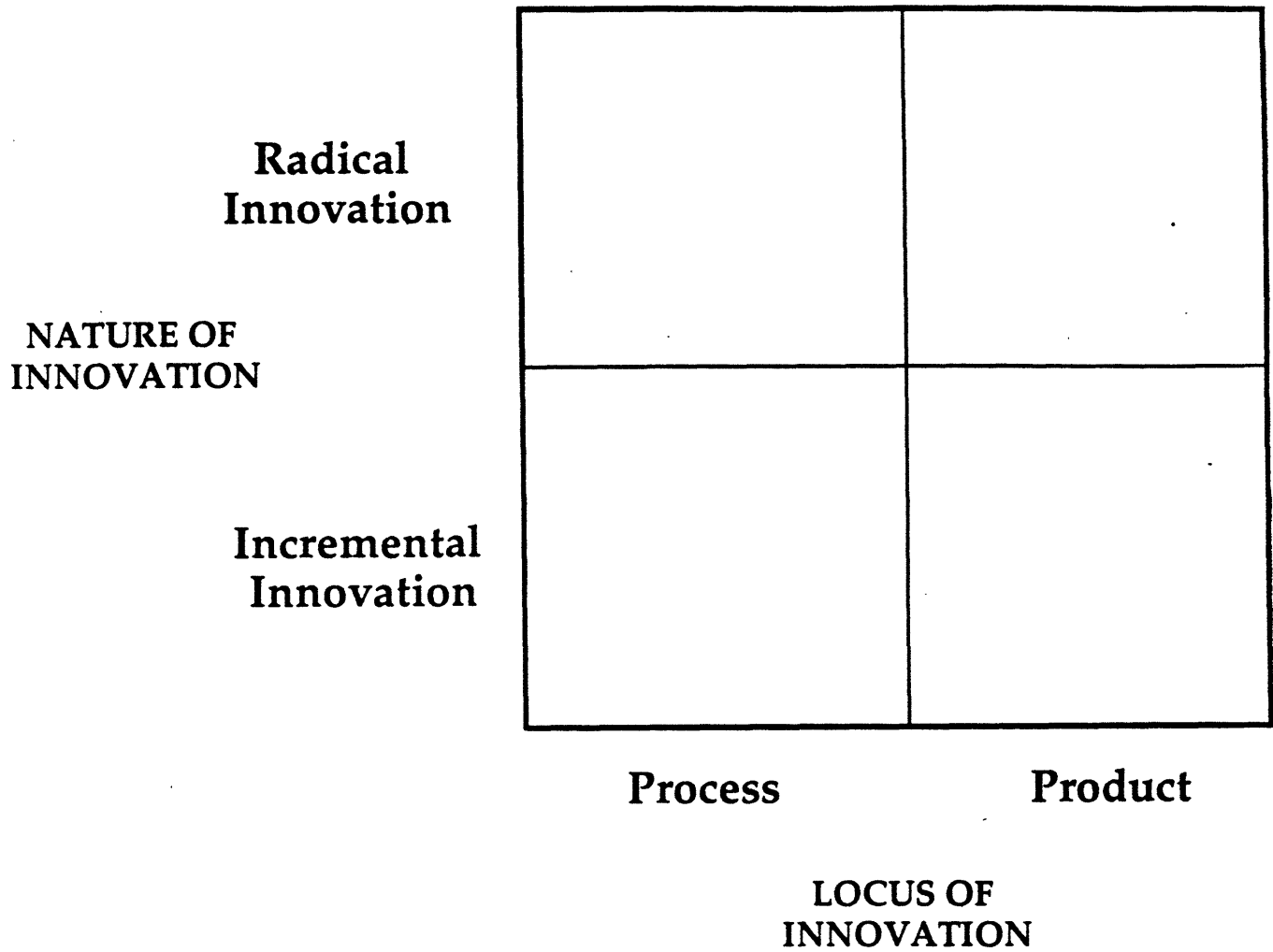


Figure 1: CASE tools as Radical and Incremental Innovations in the Process and Product of Systems Development

initial working hypothesis, I propose that the main target of CASE tools is the *process* of systems development, and that this is where we can reasonably expect to see the tools exert significant influence. Further, CASE tools have been developed and marketed with the aim of “increasing programmer productivity” and “improving the discipline of systems development.” The primary aim has been to enhance existing systems development expertise, capabilities, and controls through automated diagramming tools, data repositories, and code generators. On the basis of this, we can reasonably expect that CASE tools will represent *incremental innovations* in the existing process of systems development--supporting and extending established knowledge, structures, and practices.

On the *product* side, CASE tools appear to have much less of a direct influence--at least in their current incarnation. However, we may expect some indirect effect of CASE tools on the product of systems development, such as enhanced interface designs (because of automated aids), increased consistency (because of built-in checks and a data dictionary), and improved reliability (because of testing support). These are clearly incremental changes to the product. Other effects such as increased quality and maintainability of systems, while frequently talked about, will have to remain speculative until we are better able to define and measure quality, and have studied systems over their lifetimes.

The theoretical framework applied to the instance of CASE tools, thus, leads to a set of expectations about the nature and locus of CASE tools innovation. These can be stated in propositional form:

Proposition 1: CASE tools are associated with incremental innovations in the process of systems development, leading to incremental organizational changes in activities, control, knowledge, and productivity.

Proposition 2: CASE tools are associated with incremental innovations in the product of systems development, reflected, for example, in increased consistency, complexity, and reliability of information systems.

It is important to note that these posited relationships are not one-sided. The organizational context and individual users will play a significant role in influencing how the CASE tools are assimilated and used [see Orlikowski 1991a]. These propositions informed my field studies in which I attempted to explore the ways in which CASE tools incrementally change the process and product of systems development, and what consequences these changes have for systems development. As will become apparent, these initial expectations had to be modified to account for the data.

Research Methodology and Field Sites

I adopted the research approach of organizational ethnography [Agar 1980; Van Maanen 1979, 1988] to conduct two field studies, utilizing a range of data collection techniques including extensive interviewing and observation of participants, documentation review, social contact, and interaction with the CASE tools. The research I conducted--inductive theory building and iterative refinement of a model over time through field data--is representative of an interpretive approach to research. I did not seek deterministic relationships, and I did not try to measure cause and effect relationships. My broader research interest is in understanding the multiple, complex, and reciprocal interactions between technology and organizations [Orlikowski 1991a]. The same is true of my research into CASE technology. The nature of the model I develop is both informed by theoretical concepts and empirically grounded. It provides a vocabulary of concepts and relationships through which we can express tendencies and probabilities that help to anticipate and explain the role, use, and impacts of CASE tools in various organizational contexts.

The first field study was conducted within SoftCo, a large software consulting company. This field study was exploratory, and intended to identify and articulate the main dimensions of CASE tools interaction with the process of systems development. Five different application projects were studied, having been strategically identified to guarantee exposure to the use of CASE tools in all major phases of the systems development process (requirements analysis, conceptual design, detailed design, programming, testing, and implementation). An average of four weeks was spent on each project, observing and interviewing team members in their daily systems development work, and in their interaction with each other and the CASE tools. One hundred and twenty interviews were conducted, each lasting an average of one and a half hours. Participants in the study spanned SoftCo's hierarchic levels from the junior analysts to senior project managers, and also included some client users and managers who were participating on the projects.

The second field study was conducted within the Information Services [IS] division of ChemCo, a large chemical and petroleum products company. This field study was intended to examine and elaborate on the findings that emerged from the experiences of SoftCo. Nine different application projects were investigated within ChemCo, representing the total number of projects utilizing CASE tools. Forty interviews were conducted, each lasting an average of one hour. Participants in the study included members from all levels of ChemCo's IS Division from junior analysts to the division manager, as well as technical specialists and users assigned to the projects.

The two companies investigated and discussed here differ on a range of dimensions such as

industry, location, size, structure, and strategy. SoftCo is in the software business, developing information systems for external clients. In 1987, it employed 13,000 consultants and earned \$600 million in revenues. ChemCo is in the petroleum business, having earned \$6.3 billion in revenues in 1987. Its IS division develops and maintains information systems for internal business units, and employs about 320 people. Both SoftCo and ChemCo have, within the past five years, assimilated an integrated set of CASE tools into their systems development practices. While the CASE tools are different in their design philosophies--one being based on Structured Systems Design Approach [Yourdon and Constantine 1978] and the other on Information Engineering [Martin and Finkelstein 1982]--they are structurally and functionally similar, both being full life-cycle tools that integrate the various phases of analysis, design, coding and testing. Hence, they offer their users similar capabilities, such as design diagramming aids, screen/report design aids, data modeling facility, data dictionary, code generation, testing support, and version support.

My interest in comparing these two companies in terms of their use of CASE tools is to examine what organizational outcomes are yielded by comparable process innovations. In particular, I am interested in seeing whether and how the use of CASE tools in these two companies supports the theoretical framework and propositions developed above. If the outcomes are similar, an interesting question to explore is how the influence of the technology proved so strong as to dominate the very different social contexts of the two organizations. If the outcomes are different, it will be interesting to examine what aspects of the organizations' social contexts led to different consequences despite the similarity in technology.

Research Results

I turn now to the data that emerged from the research sites. For each site I explore three themes:

- the motivation behind the adoption of CASE tools,
- the resulting changes in process and product of systems development,
- the nature of organizational changes occasioned by use of the tools.

SoftCo

Motivation for Adoption of CASE tools

SoftCo, in operation since the sixties, develops computer-based information systems for its clients across various industries: financial services, manufacturing, retail, and government. These information systems are typically large, stand-alone, transaction-processing applications used by clients to support their major administrative activities. SoftCo's operations are organized by project, with project teams varying from around ten to over a hundred personnel, and projects

extending from a few months to a number of years in duration. Project costs range from a hundred thousand to a few million dollars. As a consequence of the growing demand for large, complex, integrated application software, SoftCo has--over the last two decades--attempted to streamline as much of its systems development practice as possible. The most recent and visible manifestation of this strategy is the construction and deployment of CASE tools within project teams.

About five years ago, SoftCo's senior managers decided that to maintain their profitability ratio and beat the competition they needed to increase productivity, hence decreasing the length of systems development, and reducing the number of consultants required on each project. They wanted to improve management leverage by increasing the number of consultants per senior manager (expanding span of control). They also wanted to diminish their dependence on the technical knowledge required for the multiple different computer configurations operated by their clients. In the past, SoftCo had to ensure that consultants knew a range of programming languages, database management systems, teleprocessing software, and operating systems to be sufficiently versatile to operate in many technological environments. Such knowledge is highly technical, idiosyncratic, and quickly becomes obsolete as new computer products continually appear on the market.

For more than a decade SoftCo's projects had used the firm's own standardized systems development methodology, which combined aspects of Yourdon structured systems design, Warnier-Orr program design, and Chen's entity-relationship data modeling. This methodology was well entrenched within the company, being thoroughly documented in fourteen volumes, taught to every employee of the firm, and its use mandated on all SoftCo projects. Senior managers attributed much of their company's success to it. Consequently, when these senior managers decided to proceed with the implementation of CASE tools, they were concerned not to lose their established expertise and experience in the methodology. As a result, they decided to build their own integrated set of CASE tools in-house, drawing heavily on their established standardized methodology and years of accumulated practice knowledge.

Changes in Systems Development Process and Product

The *process* of systems development underlying the work of SoftCo project teams was not significantly changed by the implementation of CASE tools because the tools extended the existing set of practices. The tools had been deliberately constructed to embody SoftCo's established and institutionalized systems development methodology. While new knowledge and skills were required to manipulate the tools--by system developers and technical support staff--these constituted additional skills and knowledge, not completely new competencies. A senior systems analyst at SoftCo observed:

There have been no real changes with the tools, except that we have more efficiency, more consistency, and more standardization. ... This is not strange though, as the tools have the same fundamental premises as the non-tooled environment did: procedural code, sequential development, structured design. And we built tools for these things, without trying to fundamentally change things.

CASE tools in SoftCo did not require understanding new principles of systems development, but merely the use of the tools and their assimilation into the existing systems development practice. The process of systems development was not transformed, but adapted to accommodate the tools within existing work routines.

SoftCo also did not intend their CASE tools to fundamentally transform their systems development *product*. They continued to deliver custom-built, stand-alone information systems on a project by project basis to their external clients. According to the clients and project managers interviewed, the information systems being built with CASE tools tended to be larger and more complex than would have been the case had CASE tools not been used. It appears that the availability of the enhanced process technology raises expectations about the feasibility of building more complex information systems with greater functionality. Thus, improvement in the process technology led to incremental improvements in the product technology. As Henderson and Clark [1990:11] note: "Incremental innovation refines and extends an established design. Improvement occurs in individual components [of a product], but the underlying core design concepts, and the links between them, remain the same." The primary components, linkages, design concepts, and functions present in the products emanating from SoftCo's systems development process after CASE tools were not fundamentally different to those that arose from SoftCo's earlier systems development process.

Organizational Changes occasioned by use of CASE tools

SoftCo introduced CASE tools into their systems development work by mandating their use on all large projects. Technical and supervisory skills were leveraged as the tools facilitated the abstraction of systems development knowledge from the underlying hardware and software environment, and served to monitor systems development work. Some organizational adjustments needed to be made to accommodate the presence of CASE tools on each project, but these were modifications to, not transformations of, the status quo. For example, new training programs for users of CASE tools were introduced, and technical specialists were included on projects to ensure the CASE tools' availability, reliability, and relevance. This, however, did not change the basic project structure used in SoftCo to develop information systems. Procedural changes were instituted to ensure that the tools were used when and how required, but all these changes were compatible with the existing standardized practice in force in SoftCo for at least a decade. The culture of SoftCo, likewise, was not fundamentally changed as a consequence of adopting CASE

tools. The tools were integrated into the established cultural assumptions, norms, and values, rather than challenging or dislodging them. Thus, no major organizational changes to accommodate the tools were deemed necessary or implemented.

SoftCo's senior managers did not intend CASE tools to fundamentally change the organization they had built around the production and delivery of information systems to clients. Rather, they intended their tools to create incremental improvements in systems development work. Their actions supported this strategy, and by and large, they have had some success with it, although not without some unintended consequences (explored elsewhere [Orlikowski 1989, 1991a, 1991b]).

ChemCo

Motivation for Adoption of CASE tools

ChemCo's IS division was formed within the company in the late eighties, representing a centralization of information systems resources from dispersed business functions to a separate division under the Vice-President of Technology. With this major structural change, the newly formed IS division took the opportunity to examine the role and status of information technology within ChemCo. Out of these deliberations and with input from business managers came the realization, as a senior IS manager noted, "that information technology is one of the few levers that we can pull to gain competitive advantage." Coupled with this realization came the acknowledgement that the IS division had inherited a wide diversity of technologies (hardware and software) with little or no integration, and extensive duplication of data. For example, the company was reputed to have as many as twenty product files, and--as one respondent graphically put it--"as many customer databases as Kellogs has cornflakes." A senior IS manager noted that ChemCo's information systems were hurting the company competitively: "Data was dispersed throughout the company, coding schemes were inconsistent and different, and people were drawing incorrect conclusions from poor data." An IS manager commented:

We had automated in isolation due to the natural evolution of systems in response to business demand and current technology. So we have islands of automation and a complex mosaic of bridges and much redundancy. Systems long outlive the business that sponsored them, and they fit an organization at a point in time. We need systems that are more generic and can span business area and time.

A strategy committee within the IS division was formed to look at alternative ways to deal with the "systems' mess," and after about six months of deliberations it proposed a long-term information systems strategy for the company. Members of the committee decided that a solution lay in the creation of a shared corporate data environment that would provide data effectively to the appropriate users, facilitating changes in business practices which would lead to improved

performance and competitive advantage. To implement this broad vision of a shared data environment, the IS division adopted the Information Engineering [IE] philosophy [Martin and Finkelstein 1982]. This philosophy provides a structured analysis and design methodology known as the Information Engineering Methodology, which is oriented around an organization's data. At the corporate level it provides both an integrated systems plan, and an enterprise-wide view of data from which the corporate data infrastructure--known as the Information Architecture--is constructed. At the project level it provides a detailed project view of data and specific development techniques with which to design and construct specific systems within the infrastructure. To facilitate the implementation of a standardized methodology and architecture, the committee decided to adopt a set of CASE tools which would support the change in systems development practice. An IS manager explained:

We need a shared architecture to guide us as we develop new systems, and we need a methodology and CASE tools to help this development, else we will retreat to our prior bad habits.

A senior IS manager at ChemCo provided a graphic illustration of the new world of information systems that they hoped the architecture would provide the company. It also serves as a useful means of distinguishing between radical and incremental innovation:

Let me give you an analog to how we were building systems before the architecture. We had this jigsaw of systems in the company--a jigsaw of molten steel. And when you took out a piece of the jigsaw, redesigned it, and then tried to put it back in its place in the jigsaw, lo and behold, you found that the jigsaw looked exactly the same as before. So what opportunity do you have to change the business when you have a jigsaw of molten steel? To change the jigsaw picture we need jigsaw pieces with flexible edges, which would allow us to change the way we do business.

Incremental innovation amounts to changes to the individual jigsaw pieces that leave the total picture intact. Radical innovation requires the reconfiguring of the jigsaw and its pieces, so that a different picture emerges.

Changes in Systems Development Process and Product

Before ChemCo had introduced the IE methodology and CASE tools, their systems development projects had not relied on a standardized set of practices or methodology. Rather, the systems development process was eclectic, ranging from no formal approach to one which drew on different methodologies, including some developed in-house. This state of affairs reflects the fact that systems development activities in ChemCo were distributed and relatively independent before their consolidation in 1986. The decision by the IS steering committee to adopt the IE methodology and CASE tools was the first time that a single, standardized approach to systems development was instituted within ChemCo. As a consequence, the systems developers, project managers, and

users required to use the IE methodology and CASE tools found it to be a significant departure from their prior modes of working. They had to acquire new knowledge and skills around the IE approach, hence adopting a fundamentally different design philosophy to the ones they had been implicitly or explicitly following in the past. The *process* of systems development had changed substantially--from being a somewhat loose collection of hybrid practices focused on organizational processes--to the mandatory use of a single, standardized methodology disciplined around data. One systems developer, finding the shift in focus from business processes to corporate data disconcerting, noted:

In our zeal to embrace the data-oriented approach we changed our mindset on everything that we've used till now, and labeled it poisonous. But we've just traded one evil for another. In the 60's we had messy data and messy processes. In the 70's we had messy data and structured processes. In the 80's we had structured data but we've forgotten all about structured processes.

Motivating ChemCo's decision to standardize on the IE methodology and CASE tools was the vision of an information architecture composed of subject databases and shared applications. Building and maintaining such an architecture requires cross-functional development projects, shared funding, and business unit cooperation. This form of systems development was significantly different to the stand-alone, application-oriented, functionally-driven and functionally-funded model of systems development that had served ChemCo for years. This latter model had been reflected in the structure of the IS division which had had separate departments mirroring the separate business units it served. These changes in the process of systems development thus were radical, not incremental as had been observed in SoftCo.

With respect to the *product* of systems development, ChemCo was no longer developing stand-alone, functional systems for internal business units on a project-by-project basis. Instead, a corporate information architecture had been designed, and a sequence of projects had been identified and initiated to construct this architecture. Systems were no longer "owned" by business units, but were shared, and their data modeled as part of the corporate data model. The IS division no longer responded to users' requests for systems. Rather, the IS division was taking a proactive role in dictating which systems would be built (according to the strategic plan and information architecture) and in what sequence. Thus, the primary components, linkages, design concepts, and functions present in the products emanating from ChemCo's systems development process after CASE tools were fundamentally different to those that had been developed before the use of IE methodology and CASE tools. The changes in the product of systems development were radical, not incremental as had been suggested by the experiences at SoftCo.

Organizational Changes occasioned by use of CASE tools

ChemCo intended their use of CASE tools--in conjunction with their adoption of the IE methodology and architecture concept--to transform their information systems work, and their business processes. Managers of the IS division indicated their full commitment to the new methodology and tools by making a number of structural and procedural changes. As a result of the adoption of the IE methodology and CASE tools, the IS division restructured itself, creating an architecture group (to develop the corporate-wide architecture), a shared applications support group spanning business functions (to maintain the architecture and shared applications), and a data and technology group (to manage the shared data resource and maintain the CASE tools). In addition, the separate business groups servicing business units were retained to maintain the existing stand-alone functional systems, and provide assistance with customized information retrieval for specialized functional needs. IS managers attempted to change their traditional way of developing systems by mandating the use of methodology and CASE tools on all new projects. They arranged for extensive training programs for their system developers and hired expensive consultants to help them understand the new systems development practice and to create new procedures and standards. Within ChemCo, the changes occasioned by the decision to implement CASE tools were discontinuous--not only to many of the systems developers and managers within ChemCo--but also in terms of ChemCo's established knowledge, structures, and practices. For example, project members spent a considerable time negotiating with members of other teams to resolve issues around the interpretation, definition, and handling of shared data. Such a practice was unnecessary and inconceivable in the earlier systems development era. To accomplish radical *process* innovation, IS managers have had to implement major structural, procedural, and cultural change. By and large, they have managed to do so.

IS managers also intended radical *product* innovation. Accomplishing this, however, has been much more difficult, and so far has eluded ChemCo. There has been no implementation of major structural, procedural, or cultural change by the business units. Instead, there has been considerable resistance by ChemCo business unit managers to the new product the IS division is attempting to deliver to them. An IS manager noted:

The company is not ready for this architecture. It is not willing to change business procedures, organization procedures, organization structures, and practices as a result of the architecture approach. ... A key aspect of the architecture approach is changing the business, and there is a problem because people do not want to change old familiar ways of doing things. People are uncomfortable with radical change, which is disruptive, expensive and time consuming.

Another manager commented:

Implementing a shared architecture in a decentralized environment is a real challenge. The problem is that not many pieces of the architecture have value to the business people and so they don't want to sponsor them. Middle level people tend to take a short term and narrow functional

view, and it is hard to focus them on the corporate view. ... They want immediate results and immediate returns on their investment.

Much of the resistance to the radical product innovation is due to its perceived threat to established organizational norms and practices. The notion of a corporate information architecture with shared access to data, for example, runs counter to the established “need to know” policy on information sharing. A senior business manager noted:

We used to have a closed information environment, that is, all doors are closed unless you prove otherwise. Now our policy is the reverse, all doors are open unless you justify otherwise. I think this strategy will probably fail because people don't want to share data. There is no tradition of open information in this company. ... People were successful in the past by building boxes and walls around their data and not sharing it.

Because sharing data is a basic premise of the information architecture, the reluctance of business people to buy into it has been a serious stumbling block for implementing radical product innovation. A senior IS manager noted: “This sort of paradigm shift is difficult to implement as people are reluctant to force searchlights into dark corners. It's very political.” The IS division has taken a number of proactive steps to attempt to overcome the inertia of established norms and practices, such as requiring the “open” information policy and outlawing the notion of data ownership. A manager explained that they had dropped the word “ownership” from their vocabulary when talking about data, because “It is too possessive, too explosive, and too functional.” These proactive steps, while helpful, have not been entirely successful, for they too are bound up in tradition and expectations about the appropriate actions and role of the IS division.

A senior technical specialist observed:

Here at ChemCo, IS tends to do as we're told. In the past, users have tended to get what they want. It's unusual for us to be pushing stuff. Our culture is such that on every IS' person's office are the words: “Systems do not drive the Business.”... Unless business people are forced to conform to the architecture they won't. They've had it their own way for so long they don't want to change as it is more work and more cost to them.

The business units at ChemCo are unwilling to commit to a radical change in the information product they receive, as this might imply an organizational transformation that they are not comfortable with. Their existing policies and norms do little to encourage adoption of change. One IS manager noted that: “Our reward systems are a problem too, as they do not reward sharing.” A related issue was that of funding, and who should pay for the construction of the architecture and the shared applications. Shared applications are currently being developed by the architecture group. However, the projects are being paid for by individual business areas. It is unclear who will retain responsibility and ownership of these systems once they are in production and being used by multiple business areas. A project manager explained:

In the past business people used to ‘pay for functionality,’ now we want them to ‘pay for data,’

and that's a hard sell because it is an enterprise need not a narrow functional need. ... The shared data environment has had an impact on the ownership of systems and data. We used to say, 'who owns the systems, owns the data,' so there was no issue in the past about which VP was on the hook for the data. That is, data followed the system and the person who paid for the system owned it, and hence also owned the data. Now data is reversed. It's not a hierarchy, but a network and now data ownership is shared across the corporation. It's no longer bounded by functional business areas. So who owns an application, who maintains it, and who changes it, is up for grabs. Data and applications no longer map to each other.

Partly as a consequence of the reluctance of business areas to commit to the radical product innovation, the IS division has not been able to fully implement the vision of their innovation.

Interpretation of Research Results

Figure 2 depicts the experiences SoftCo and ChemCo have had with their CASE tools. The differences in organizational outcomes can be seen to be linked to the different social contexts (structures, practices, knowledge, and culture) and managerial strategies within which the CASE tools were adopted. The changes in process and product associated with CASE tools have been mapped against the type of innovation observed (incremental or radical). SoftCo treated its CASE tools as an incremental innovation in the process of systems development, and used the tools to effect incremental improvements in the product of its work. ChemCo, on the other hand, used the CASE tools to usher in a new approach to systems development, significantly changing the process by which information systems are built. In addition, IS managers in ChemCo are fundamentally changing--not without some difficulty--the nature of the product they deliver to their clients, from stand-alone systems to an integrated architecture and shared applications.

It is critical that this framework not be misinterpreted. I am not suggesting that CASE tools caused incremental innovations in SoftCo's process and product, and radical innovations in ChemCo's process and product. Rather, I am suggesting that in order to understand the sorts of changes that are associated with CASE tools, it is necessary to understand the strategic intentions and social context within which they were adopted and are being used. This, however, is not, a one-sided relationship. I see the relationship between this context and the CASE tools to be one of reciprocal interaction (see Orlikowski [1991a] for a detailed exposition of this relationship). Senior managers' intentions and the social context lead to the adoption of a certain set of CASE tools, and influence the conditions under which these tools are used--what Bjørn-Andersen, Eason and Robey [1986] refer to as "planned impact" and "contingent impact" respectively. Use of the tools in turn, leads to certain changes to the existing social context (potentially changing, disrupting, or reaffirming it over time) and may even encourage changes in managers' strategies. These latter are often unanticipated consequences of the adoption of technology [Orlikowski 1991a].

NATURE OF INNOVATION Radical Innovation	<i>ChemCo</i>	<i>ChemCo</i>
	<i>SoftCo</i>	<i>SoftCo</i>
Incremental Innovation	Process	Product
	LOCUS OF INNOVATION	

Figure 2: CASE tools as Incremental Innovations in SoftCo and Radical Innovations in ChemCo

Thus, in the case of SoftCo, CASE tools were designed and deployed to leverage existing knowledge and practices. The intention of senior managers was to institute productivity improvements in the process of systems development. In the case of ChemCo, on the other hand, CASE tools were acquired to facilitate the implementation of a new information systems strategy, and to construct a corporate-wide information architecture. The intention of senior managers was to create radical changes in both the process and product of systems development. This framework suggests that the organizational changes following the adoption of CASE tools in organizations must be understood and evaluated in the light of strategic intentions and social contexts.

Extending the Theoretical Model

While the data from the first field study would support the initial research propositions stated above, analysis of the data from the second field study has disconfirmed the research propositions. Rather than representing incremental innovations in process and product, CASE tools appear to represent different innovations depending on the motivation behind their employment, and the social context of the organization and information systems unit. The data suggest different organizational consequences from apparently similar process innovations. This finding resembles that of Barley [1986, 1990] in which he found different organizational changes resulted from similar technological innovations. Discriminating the nature of innovation attempted and its locus of influence (as depicted in Figure 2) allows us to explain why two companies--apparently attempting the same process innovation (integrated CASE tools)--should experience such different outcomes. These variations allow the construction of a fuller, grounded theory explanation that accounts for the differences observed across the two organizations' use of CASE tools.

It is important to note that the existing structures, practices, and culture can be both facilitators or inhibitors of innovations depending on the type of innovation attempted. Because the CASE tools introduced to SoftCo were based on and reaffirmed the existing, standardized systems development methodology, the established structures, practices, and culture enabled the adoption of the CASE tools. That is, the established and deeply ingrained methodology of systems development in force within SoftCo served to facilitate and greatly expedite the diffusion and implementation of an incremental innovation. However, if SoftCo's managers had attempted to introduce radical change in their systems development process, and had deliberately built or acquired CASE tools that challenged the existing, standardized methodology, SoftCo's established structures, practices, and culture would have constrained adoption of the CASE tools.

The findings of these two research studies make it clear that we need to modify the original research propositions presented earlier. In particular, we need to take into account the motivation

behind the adoption of CASE tools, and the existing social context in the organization. As a first statement of the phenomenon, the following propositions provide some useful initial insights for researchers and practitioners into the consequences of adopting CASE tools in organizations.

Proposition 1: When the motivation behind the adoption of CASE tools is to augment the existing knowledge and activities of systems development, they constitute incremental innovations in process, and hence will not require major organizational transformations in the structure, practice, and culture of the information systems unit.

Proposition 2: When the motivation behind the adoption of CASE tools is to augment the features and functionality of information systems being delivered to users, they constitute incremental innovations in product, and hence will not require major organizational transformations in the structure, practice, and culture of the business units using the information systems.

Proposition 3: When the motivation behind the adoption of CASE tools is to facilitate discontinuous changes in the knowledge and activities of systems development, they constitute radical innovations in process, and hence will require major organizational transformations in the structure, practice, and culture of the information systems unit.

Proposition 3a: Where the existing knowledge and activities of systems development are deeply held and institutionalized within an organization, adoption of a radical innovation in process by the information systems unit will be difficult.

Proposition 4: When the motivation behind the adoption of CASE tools is to facilitate discontinuous changes in the form and functionality of information systems being delivered to users, they constitute radical innovations in product, and hence will require major organizational transformations in the structure, practice, and culture of the business units using the information systems.

Proposition 4a: Where the existing product of systems development is deeply held and institutionalized within an organization, adoption of a radical innovation in product by business units will be difficult.

From a managerial perspective, proposition 1 suggests that little structural, procedural, or cultural changes will be needed to accommodate an incremental process innovation. And so it was at SoftCo, which was able to use CASE tools to leverage established practices, knowledge, and structures, without major organizational change. Proposition 2 suggests that incremental product innovation requires incremental not radical changes in structure, practices and culture. The data imply that SoftCo delivered information systems which differed from their predecessors in degree not kind. These are changes which we would expect to have incremental implications for the receiving business units.

With respect to radical process innovation and propositions 3 and 4, the case of ChemCo clearly bears out the expectation that for such a change to take place, the information systems division must foster significant structural, procedural, and cultural change, not only in its own unit but throughout the organization. Without such a reorganization and reframing, the paradigm shift implied in a radical change will not be accomplished. Where the existing process of systems development is deeply entrenched radical change will be more difficult to implement. Because there was no standardized, institutionalized process of systems development within ChemCo at the time of CASE tools adoption, the radical process change was adopted without too much resistance. With respect to radical product innovation, ChemCo's business units have refused to commit to the organizational transformations required of them to benefit from the significant change in their information systems product. Because the existing systems development product (stand-alone applications for separate business units) is deeply entrenched in the decentralized culture of ChemCo, radical change to a corporate, shared information architecture has proven to be a formidable challenge. As a result, no successful product innovation has occurred to date, and much ambiguity, turf conflict, and resistance plagues the organization. While this data does not show that radical structural and cultural change by business units will lead to successful adoption by the business users of the radical product innovation, it does show that where such change is lacking there is no adoption of the radical innovation.

In considering the difference between the process and the product of systems development it becomes clear that different players are more intimately involved in these different aspects of information systems. That is, systems developers and managers of information systems units are fully involved in the *process* of systems development, and hence any innovations in this process will affect them most directly. As primary users of the process, the IS work practices, structures, and skills will probably require modification to accommodate the process innovation. Such innovations are relatively contained within the information systems unit. Where there is also an innovation in the *product* of systems development, however, things are not so contained. While systems developers and managers of the information systems unit are intimately implicated in the product being developed, the clients to whom the product will be delivered are also directly concerned with it. As primary users of the product, the business users' work practices, structures, and skills will probably require modification to accommodate the product innovation. Thus, any innovation in the product of systems development affects not only the information systems unit and its workers, but also its clients--the business users, their managers, and their unit.

Conclusions

To conclude, I will briefly examine the implications of the model and the research findings for systems development research and practice.

Implications for Research

This paper has presented a theoretical model for understanding and managing the adoption of CASE tools and its influence on systems development. While more empirical work is clearly needed to elaborate and verify the model and propositions developed here, a useful starting point has been made. The theoretical model suggests that when thinking about the adoption of CASE tools and its influence on systems development, it is necessary to understand the history and motivation behind the adoption of the innovation (objectives, strategic intentions, and scope of the innovation) and the social context into which the innovation is deployed (structure, practices, and culture). This understanding allows us to better explain and predict the influence of CASE tools, as it suggests where the influence of the CASE tools innovation is likely to be, and further, what challenges the innovation is likely to represent to the status quo.

Empirical validation and elaboration of these propositions in other settings is clearly needed. First, it is necessary to examine other organizations that attempted radical innovation in process and product, and those that attempted incremental innovation in both. Second, other positions on the grid of Figure 1 would also be usefully studied to determine the organizational consequences of mixed innovations. Both SoftCo and ChemCo adopted systems development innovations that were consistent across process and product, that is, either both radical or both incremental innovations. It is however, conceivable to think of other combinations, and Figure 1 suggests that other possible adoption outcomes are possible: (i) where CASE tools are associated with radical changes in the process of systems development, and only incremental changes in the product delivered to clients; and (ii) where CASE tools are associated with incremental changes in the process of systems development, and radical changes in the product delivered to clients. Further field studies are in order to further understand the implications of these different combinations of innovations.

This paper has only focused on two factors affecting the implementation of an innovation. While I believe these factors are very important, there are also other important factors that may influence innovation adoptions, for example, human resource policies, individual differences among system developers, training strategies, and environmental conditions (see Kwon and Zmud [1987], Kazanjian and Drazin [1986], Myers and Goes [1988], and Rousseau [1989] for a discussion of some of these factors).

Implications for Practice

The theoretical model developed and presented here has significant implications for managerial action. As was stated earlier--because it requires departures from existing assumptions, norms, knowledge, structures, and practices--implementing radical innovation is significantly more difficult than implementing incremental innovation. Stability and equilibrium are beneficial to organizations in that they allow goals to be pursued and work to be accomplished. At the same time, the apparent comfort and utility of an existing approach--however imperfect--often discourages attempts to institute unknown and potentially uncomfortable, disruptive, or risky changes. This is where strong leadership and commitment--both symbolic and substantive--are critical to making the new order not only less risky, but preferred.

In the light of the model we can expect that for an information systems unit to successfully implement a **radical process innovation** through CASE tools, senior information systems managers need to institute significant organizational changes within the information systems unit. Apart from changes to the information systems' unit structure, practices, and resource allocation schemes, changes in systems developers' perspectives and orientations are critical for obtaining their commitment to and understanding of the desired new order. Without these fundamental changes in assumptions, knowledge, and values, there will be no change or no sustained change in individuals' actions, and ultimately no enactment of the infrastructural changes. For successful **incremental process innovation** we would not expect to see the same level of structural and cultural transformation. Some retraining and procedural modifications are likely to be needed, but these will not constitute fundamental changes to the existing practice. Rather, incremental innovations reaffirm and advance the status quo, they do not overturn it.

When it comes to **radical product innovation**, however, it is the senior and middle managers of the business units that need to institute significant substantive and symbolic organizational changes. As **primary users** of the innovation--the radical new product--these users need to transform their assumptions, knowledge, and infrastructure to take advantage of the radical new information systems. To the extent that business managers do not internalize the paradigm shift implicit in a radical product innovation--as was evident in the case of ChemCo--the influence of the CASE tools innovation will be limited to the information systems department and systems development process. Not only will this reduce the potential impact of the innovation, it will likely cause tension, frustration, and some hostility between the business users and the information systems developers. It will also undermine efforts by the organization to reap the expected benefits from the technology implementation.

As noted earlier, radical or discontinuous change is, by definition, difficult to accomplish. There are many organizational barriers to radical or discontinuous change [Gersick 1991; Pennings 1988; Tushman and Romanelli 1985] which need to be overcome for successful adoption of a radical innovation. Some areas where managerial action may be targeted include: mental models, human resource strategies, resource allocation policies, and organizational or unit structure and procedures.

- Individuals' mental models tend to be oriented around established practices and norms, and may limit perception and understanding of an innovation, or even the need for it. Reorientation through workshops and training may help to break old assumptions and perceptions.
- Disincentives to change such as anxiety, fear, job insecurity, and uncertainty often encourage avoidance of and resistance to radical organizational change. Creating effective human resource strategies that accommodate changed skills, roles, knowledge, responsibilities, and alternative career paths in line with the new order may ease the transition. Use of opinion leaders to act as role models for the more cautious users may emphasize and personalize the benefits in a way that public relations cannot.
- An existing organization has much inertia built into its structures, policies, and procedures, which represent expectations, obligations, resource commitments, and dependencies. These vestiges of the old order serve as barriers to the adoption and diffusion of new ideas and practices. Managers need to clearly signify and implement the new order by actively establishing new policies, reorganizing structurally, and changing obsolete procedures or inappropriate norms.

A major implication of the model developed in this paper is that it is essential to understand the motivation and context of the innovation in order to be able to successfully implement it. Understanding the locus of the innovation--whether the process or product (or both) of systems development is affected--and its nature relative to the existing social context--incremental or radical--provides insights into what organizational changes are likely, and how they might be managed.

Acknowledgements

Grateful thanks are due to Debra Hofman for many helpful comments and discussions. This research was supported by the research program on "Improving Systems Delivery" at the MIT Center for Information Systems Research.

References

- Barley, S. (1986) "Technology as an Occasion for Structuring: Evidence from Observations of CT Scanners and the Social Order of Radiology Departments," *Administrative Science Quarterly*, 31, 78-108.
- _____ (1990) "The Alignment of Technology and Structure through Roles and Networks," *Administrative Science Quarterly*, 35, 61-103.
- Bayer Judy and Melone, Nancy (1989) "A Critique of Diffusion Theory as a Managerial Framework for Understanding Adoption of Software Engineering Innovations," *Journal of Systems and Software*, 9, 161-166.
- Bjørn-Andersen, Niels, Eason, Ken and Robey, Daniel (1986) *Managing Computer Impact: An International Study of Management and Organizations* Norwood NJ: Ablex Publishing.
- Brancheau, J.C. and Wetherbe, J.C. (1990) "The Adoption of Spreadsheet Software: Testing Innovation Diffusion Theory in the Context of End-User Computing," *Information Systems Research*, 1, 2, 115-143.
- Cooper, R.B. and Zmud, R.W. (1990) "Information Technology Implementation Research: A Technological Diffusion Approach," *Management Science*, 36, 2, 123-139.
- Dewar, R.D. and J.E. Dutton (1986) "The Adoption of Radical and Incremental Innovations: An Empirical Analysis," *Management Science*, 32, 11, 1422-1433.
- Ettlie, J.E., W.P. Bridges and R.D. O'Keefe (1984) "Organization Strategy and Structural Differences for Radical versus Incremental Innovation," *Management Science*, 30, 6, 682-695.
- Henderson, R.M. and K.B. Clark (1990) "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms," *Administrative Science Quarterly*, 35, 9-30.
- Iacono, Suzanne and Kling, Rob (1988) "Computer Systems as Institutions: Social Dimensions of Computing in Organizations," *Procs. of the Ninth International Conference on Information Systems*, Minneapolis MN.
- Kazanjian, Robert K. and Drazin, Robert (1986) "Implementing Manufacturing Innovations: Critical Choices of Structure and Staffing Roles," *Human Resource Management*, 25, 3, 385-403.
- Kwon, Tae H. and Zmud, Robert W. (1987) "Unifying the Fragmented Models of Information Systems Implementation," in R.J. Boland and R.A. Hirschheim *Critical Issues in Information Systems Research*, New York: John Wiley & Sons, 227-251.
- Leonard-Barton, Dorothy A. (1988) "Implementation as Mutual Adaptation of Technology and Organization," *Research Policy*, 17, 251-267.
- Leonard-Barton, Dorothy A. and Deschamps, Isabelle (1988) "Managerial Influence in the Implementation of New Technology," *Management Science*, 34, 10, 1252-1265.
- Markus, M. Lynne (1983) "Power, Politics, and MIS Implementation." *Communications of the ACM*, 26, 430-444.

Martin, J. and M. Finkelstein (1982) *Principles of Information Engineering* Englewood Cliffs, NJ: Prentice-Hall.

Meyerson, D. and J. Martin (1987) "Cultural Change: An Integration of Three Different Views," *Journal of Management Studies*, 24, 6, 623-647.

Moore, Gary C. and Benbasat, Izak (1989) "Development of an Instrument to Measure the Perceived Characteristics of Adopting an Information Technology Innovation," Working Paper No. 89-05 University of Calgary, Alberta, Canada.

Myers, Alan D. and Goes, James B. (1988) "Organizational Assimilation of Innovations: A Multilevel Contextual Analysis," *Academy of Management Journal*, 31, 4, 897-923.

Norman, R.J. and Nunamaker, J.F. (1989) "CASE Productivity Perceptions of Software Engineering Professionals," *Communications of the ACM*, 32, 9, 1102-1108.

Orlikowski, Wanda J. (1988) "Information Technology and Post-Industrial Organizations: An Exploration of the Computer-Mediation of Production Work," Unpublished Doctoral Dissertation, Stern School of Business, New York University, New York.

_____ (1989) "Division among the Ranks: The Social Implications of CASE tools for Systems Developers" *Proceedings of the Tenth International Conference on Information Systems*, Boston MA, December.

_____ (1991a) "The Duality of Technology: Rethinking the Concept of Technology in Organizations" Forthcoming in *Organization Science*.

_____ (1991b) "Integrated Information Environment or Matrix of Control?: The Contradictory Implications of Information Technology" Forthcoming in *Accounting, Management and Information Technologies*.

Pennings, Johannes (1988) "Information Technology in Production Organizations," *International Studies of Management and Organization*, XVII, 4, 68-89.

Rice, Ronald E. and Rogers, Everett M. (1980) "Reinvention in the Innovation Process," *Knowledge*, 1, 4, 499-514.

Rogers, Everett M. (1983) *Diffusion of Innovations*, (3rd. ed.), New York: The Free Press.

Rousseau, Denise M. (1989) "Managing the Change to an Automated Office: Lessons from Five Case Studies," *Office: Technology and People*, 4, 31-52.

Starbuck, William H. (1983) "Organizations as Action Generators," *American Sociological Review*, 48, 91-102.

Tushman, M.L. and E. Romanelli (1985) "Organizational Evolution: A Metamorphosis Model of Convergence and Reorientation," *Research in Organizational Behavior*, 7, 171-222.

Utterback, J.M. and Abernathy, W.J. (1975) "A Dynamic Model of Process and Product Innovation," *Omega*, 3,6, 639-656.

Yourdon, E. and Constantine, L. L. (1978) *Structured Design* New York: Yourdon Press.

Zmud, R.W. (1984) "An Examination of 'Push-Pull' Theory Applied to Process Innovation in Knowledge Work," *Management Science*, 30, 6, 727-738.