

# CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development

**By: Wanda J. Orlikowski**  
Sloan School of Management  
Massachusetts Institute of Technology  
50 Memorial Drive (E53-329)  
Cambridge, Massachusetts 02139  
U.S.A.

## Abstract

*This paper presents the findings of an empirical study into two organizations' experiences with the adoption and use of CASE tools over time. Using a grounded theory research approach, the study characterizes the organizations' experiences in terms of processes of incremental or radical organizational change. These findings are used to develop a theoretical framework for conceptualizing the organizational issues around the adoption and use of these tools—issues that have been largely missing from contemporary discussions of CASE tools. The paper thus has important implications for research and practice. Specifically, the framework and findings suggest that in order to account for the experiences and outcomes associated with CASE tools, researchers should consider the social context of systems development, the intentions and actions of key players, and the implementation process followed by the organization. Similarly, the paper suggests that practitioners will be better able to manage their organizations' experiences with CASE tools if they understand that such implementations involve a process of organizational change over time and not merely the installation of a new technology.*

**Keywords:** CASE tools, change management, organizational change, systems development, systems implementation

**ISRL Categories:** AF10, AI0103, AI0108, FA07, FC04, FD05

## Introduction

CASE (computer-aided software engineering) tools have generated much interest among researchers and practitioners as potential means for easing the software development and maintenance burden threatening to overwhelm information systems (IS) departments. While interest and investment in CASE tools have been rising steadily, actual experiences with tools have exhibited more ambiguity. For example, while some studies report improvements in productivity from the use of CASE tools (Banker and Kauffman, 1991; Necco, et al., 1989; Norman and Nunamaker, 1988; Swanson, et al., 1991), others find that the expected productivity gains are elusive (Card, et al., 1987; Yellen, 1990) or eclipsed by lack of adequate training and experience, developer resistance, and increased design and testing time (Norman, et al., 1989; Orlikowski, 1988b; 1989; Vessey, et al., 1992).

While these contradictory experiences with CASE tools appear puzzling and difficult to interpret, the research presented in this paper suggests that by shifting the focus away from specific outcome expectations, we may be able to make some sense of the apparently inconsistent findings. This paper argues that the adoption and use of CASE tools should be conceptualized as a form of *organizational change* and that such a perspective allows us to anticipate, explain, and evaluate different experiences and consequences following the introduction of CASE tools in organizations.

To date, there has been no systematic examination or formulation of the organizational changes surrounding CASE tools. Much of the literature on CASE tools has tended to focus on discrete outcomes, such as productivity, systems quality, and development costs, while neglecting the intentions and actions of key players, the process by which CASE tools are adopted and used, and the organizational context within which such events occur. Issues of intentions, actions, pro-

cess, and context around information technology are not new to the IS field. For example, implementation research has looked at the process through which technology is introduced (Ginzberg, 1981; Rogers, 1983). Also, the interactionist approach (Markus, 1983) and the reinforcement politics approach (George and King, 1991) have examined the role of social context in shaping the introduction and use of technology, while the structuration perspective (DeSanctis and Poole, forthcoming; Orlikowski and Robey, 1991) has emphasized the centrality of players' deliberate, knowledgeable, and reflective action in shaping and appropriating technology. Yet contemporary discussions around CASE tools in research, education, and practice tend to gloss over these issues.

In this paper, the implementation of CASE tools is understood as a specific case of technology-based organizational change. As such, the core research question is: *What are the critical elements that shape the organizational changes associated with the adoption and use of CASE tools?* In answering this question, the paper is organized as follows. First is a description of the empirical findings that emerged from a grounded theory study of two organizations that implemented CASE tools in their systems development operations. Next is the development of a theoretical framework that conceptualizes the findings in terms of three central categories: strategic conduct, institutional context, and change process.

The grounded theory approach was useful here because it allows a focus on contextual and processual elements as well as the action of key players associated with organizational change—elements that are often omitted in IS studies that rely on variance models and cross-sectional, quantitative data (Markus and Robey, 1988; Orlikowski and Baroudi, 1991). While the findings of this grounded theory study are detailed and particularistic, a more general explanation can also be produced from the results (Dutton and Dukerich, 1991; Eisenhardt, 1989; Leonard-Barton, 1990). Yin (1989a) refers to this technique as "analytic generalization" to distinguish it from the more typical statistical generalization that generalizes from a sample to a population. Here the generalization is of theoretical concepts and patterns. This generalization is further extended in this paper by combining the inductive concepts

generated by the field study with insights from existing formal theory, in this case from the innovation literature (a strategy recommended by Glaser and Strauss, 1967). The outcome is a general conceptualization of the organizational changes associated with adopting and using CASE tools that should both contribute to our research knowledge and inform IS practice.

The paper makes three principal contributions. First, drawing on the rich data of two organizations' experiences, the paper generates a grounded understanding of the changes associated with implementing CASE tools in systems development. This grounded theory is valid empirically "because the theory-building process is so intimately tied with evidence that it is very likely that the resultant theory will be consistent with empirical observation" (Eisenhardt, 1989, p. 547). While many believe that building theory from a limited number of cases is susceptible to researchers' preconceptions, Eisenhardt (1989) argues persuasively that the opposite is true. The iterative comparison across sites, methods, evidence, and literature that characterizes such research leads to a "constant juxtaposition of conflicting realities [that] tends to 'unfreeze' thinking, and so the process has the potential to generate theory with less researcher bias than theory built from incremental studies or armchair, axiomatic deduction" (p. 546). Second, the grounded theory developed in this paper adds substantive content to our understanding of the central role played by individual actors, their institutional context, and the processes they enact in adopting and using CASE tools. Such an understanding has been absent from the research and practice discourses on CASE tools. The approach followed here focuses specifically on developing such an understanding, thus bringing a fresh set of issues to the already researched topic of CASE tools (Eisenhardt, 1989). Third, the paper integrates a specific grounded theory with the more formal insights available from the innovation literature, developing a more general framework that will allow researchers and practitioners to explain, anticipate, and evaluate various organizational changes associated with the adoption and use of CASE tools.

The paper is structured as follows. The first section describes the research methodology and the two research sites. The next section presents the

research findings, describing the experiences of each organization in turn. The discussion section follows, integrating the specific concepts and findings of the field research with insights from the innovation literature into an analytic framework for conceptualizing CASE tools adoption and use in organizations. The conclusion then assesses the contribution of the research framework and findings, both for future research and for the management of CASE tools in organizations.

## Research Methodology

The research methodology followed is that of grounded theory (Glaser and Strauss, 1967; Martin and Turner, 1986; Turner, 1983), with an aim of generating a descriptive and explanatory theory of the organizational changes associated with CASE tools rooted in the experiences of specific systems development operations. This approach has been effectively used in organizational research (Ancona, 1990; Elsbach and Sutton, 1992; Isabella, 1990; Kahn, 1990; Pettigrew, 1990; Sutton, 1987) and is adopted here for three primary reasons.

First, grounded theory "is an inductive, theory-discovery methodology that allows the researcher to develop a theoretical account of the general features of a topic while simultaneously grounding the account in empirical observations or data" (Martin and Turner, 1986, p. 141). This generative approach seemed particularly useful here given that no change theory of CASE tools adoption and use has been established to date. While models of information technology implementation do exist (Ginzberg, 1981; Lucas, 1978; Markus, 1983) these deal largely with the development stages of IS implementation and focus extensively on user involvement and user relations. As a result, they are less applicable to the issue of organizational change in general and to the case of CASE tools adoption and use in particular.

Second, a major premise of grounded theory is that to produce accurate and useful results, the complexities of the organizational context have to be incorporated into an understanding of the phenomenon, rather than be simplified or ignored (Martin and Turner, 1986; Pettigrew, 1990). As indicated above, a number of theoretical approaches emphasize the criticality of organiza-

tional context in shaping technology use in organizations. Such a conviction also informs the research in this paper, and the use of a grounded theory methodology allows the inclusion and investigation of this key organizational element.

Third, grounded theory facilitates "the generation of theories of process, sequence, and change pertaining to organizations, positions, and social interaction" (Glaser and Strauss, 1967, p. 114). As indicated above, the change an organization undergoes in adopting and assimilating CASE tools, as well as the processes of appropriation and use that system developers engage in to incorporate the tools in their work lives, has tended to be neglected in the CASE tools literature. A research approach that specifically includes elements of process and change was thus particularly appropriate here.

These three characteristics of grounded theory—inductive, contextual, and processual—fit with the interpretive rather than positivist orientation of this research. The focus here is on developing a context-based, process-oriented description and explanation of the phenomenon, rather than an objective, static description expressed strictly in terms of causality (Boland, 1979; 1985; Chua, 1986; Orlikowski and Baroudi, 1991). In the language of Markus and Robey (1988) and Mohr (1982), the paper develops a process, not a variance theory. Such a theory describes and explains the process of adopting and using CASE tools in terms of an interaction of contextual conditions, actions, and consequences, rather than explains variance using independent and dependent variables (Elsbach and Sutton, 1992). This orientation "gives primacy to realism of context and theoretical and conceptual development as research goals" (Pettigrew, 1990, p. 283).

The methodology of grounded theory is iterative, requiring a steady movement between concept and data, as well as comparative, requiring a constant comparison across types of evidence to control the conceptual level and scope of the emerging theory. As Pettigrew (1989) notes, this "provides an opportunity to examine continuous processes in context in order to draw out the significance of various levels of analysis and thereby reveal the multiple sources of loops of causation and connectivity so crucial to identifying and explaining patterns in the process of

change" (p. 14). To facilitate this iteration and comparison, two field sites were studied and analyzed in turn, a strategy also adopted by Kahn (1990). The initial concepts thus emerged in one organizational context and were then contrasted, elaborated, and qualified in the other.

### Site selection

Following Glaser and Strauss' (1967) technique of theoretical sampling, the two organizations were selected for their similarities as well as their differences. Theoretical sampling requires paying attention to theoretical relevance and purpose. With respect to relevance, this selection process ensures that the substantive area addressed—here the adoption and use of CASE tools—is kept similar, or as Eisenhardt (1989) notes, "is likely to replicate or extend the emergent theory" (p. 537). Thus, both organizations chosen for this study had within the past few years implemented CASE tools into their systems development operations and mandated their use on all new systems development work. In addition, the CASE tools themselves, while not identical, were compatible across the two organizations in that both were life-cycle tools that integrate the phases of analysis, design, coding, and testing.<sup>1</sup> Both sets of tools provide similar capabilities, such as design aids (e.g., data flow diagrams), data modeling facilities (e.g., entity-relationship modeling), screen and report design utilities, data repositories, code generators, test data generation, and version control. While the one set of CASE tools is based on the structured systems design approach (Yourdon and Constantine, 1978), the other is based on information engineering (Martin, 1990a; 1990b; 1990c). Thus, the two sets of CASE tools are philosophically similar, drawing on the same basic software engineering tenets of functional decomposition, separation of process and data, and sequential development phases.<sup>2</sup>

Because the purpose of the research was to generate theory applicable to various organizational contexts and distinguish different

change processes, differences were sought in organizational conditions, such as the nature and scope of systems development activity and the method of CASE tools acquisition. As a result, the two companies selected also differ on other organizational dimensions, such as industry, location, size, structure, and culture. The consulting firm (SCC)<sup>3</sup> is in the software business, developing information systems for external clients. In 1987, it employed 13,000 consultants and earned \$600 million in revenues. SCC acquired its CASE tools by developing them in-house. The petro-chemical firm (PCC) is in the petroleum products business, having earned \$6.3 billion in revenues in 1987. It has an internal information systems division, which employs 320 people and develops and maintains information systems for internal business units. PCC acquired its CASE tools by purchasing them from an outside vendor. These differences in organizational conditions allowed useful contrasts to be made during data analysis, which challenged and elaborated the emerging concepts.

### Data sources

In both research sites, data were collected through a variety of methods: unstructured and semi-structured interviewing, documentation review, and observation. This triangulation across various techniques of data collection is particularly beneficial in theory generation because it provides multiple perspectives on an issue, supplies more information on emerging concepts, allows for cross-checking, and yields stronger substantiation of constructs (Eisenhardt, 1989; Glaser and Strauss, 1967; Pettigrew, 1990).

Data collection focused on the topics of context, technology, key players, and change process. It sought information on, among other things: the environment, mission, structure, and culture of the firm and IS department; the size, location, and composition of the IS department; the history of systems development in the firm (including configurations, standards, policies, and procedures); the rationale for the tools' acquisition; the nature and methodology of the tools acquired; the CASE tools' implementation strategy and schedule; managerial commitment; user involvement; training; individual and team experiences with the

<sup>1</sup> Such tools are also known as "integrated CASE tools" or ICASE (Banker and Kauffman, 1991).

<sup>2</sup> In contrast, for example, to more recent and significantly different systems development philosophies, such as object-oriented development (Fichman and Kemerer, 1993).

<sup>3</sup> The names of the two organizations have been disguised.

CASE tool; and changes associated with use of the CASE tools in such areas as skills, knowledge, work, performance, and interaction with peers, superiors, and users.

Data collection, coding, and analysis proceeded iteratively (Glaser and Strauss, 1967) with the early stages of the research being more open-ended, and later stages being directed by the emerging concepts, and hence involving more strategic selection of informants and more structured interview protocols.<sup>4</sup> Overlapping data analysis with data collection, as Eisenhardt (1989) notes, provides a number of advantages: "[It] not only gives the researcher a head start in analysis, but more importantly allows researchers to take advantage of flexible data collection. Indeed, a key feature of theory-building case research is the freedom to make adjustments during the data collection process" (p. 539).

The first field study was conducted within a large, multinational software consulting firm (SCC), which builds customized software applications for client firms across various industries, such as financial services, manufacturing, retail, and government. The data were collected as part of a larger research study that consisted of an in-depth field study conducted over eight months in three SCC offices (Orlikowski, 1988a).<sup>5</sup> Five different application projects were studied, having been strategically selected to allow 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 nineteen interviews were

conducted, each lasting an average of one and a half hours. Participants in the study spanned SCC's hierarchic levels, from junior analysts to senior project managers, and included some client users who were participating in the projects. Table 1 shows the type and amount of interviews conducted at SCC.

The second field study was conducted within an information services (IS) division, located at the headquarters of a large petroleum products company (PCC). Nine different application projects were examined, representing the total number of PCC projects utilizing CASE tools. Forty interviews were conducted, each lasting an average of one hour. Participants in the study included members at multiple levels of PCC's IS division, from junior analysts to the division manager, including technical support specialists and a few of the users assigned to the projects. Informal discussions were also conducted with the consultants and vendor specialists that were facilitating PCC's adoption and use of the CASE tools. Table 2 shows the type and amount of interviews conducted at PCC.

In each site, documentation about the company, the IS projects, and the CASE tools were examined. Use of the particular tools at each site was observed, both in "observe-only" mode and in a mode that had the system developers "talk through" their various systems development tasks. While the primary unit of analysis was the organization or organizational department undergoing changes associated with CASE tools, the grounded theory approach encourages the collection of inter-related data at other levels of analysis (Pettigrew, 1990; Yin, 1989a; 1989b). In this case, the experiences of individual developers, technical specialists, managers, and users, as well as the larger institutional context, were also examined. As Leonard-Barton (1990) notes about her studies of technological transfer, "In order to understand all the interacting factors . . . it was necessary that the research methodology slice vertically through the organization, obtaining data from multiple levels and perspectives" (p. 249).

### *Data analysis*

The data were analyzed within each site, as well as across the two sites, to detect similarities and compare differences. Within SCC (the first site),

<sup>4</sup> Copies of the more structured interview protocols are available from the author on request.

<sup>5</sup> This larger study examined the social process of developing, implementing, and using process technology in a software production environment. Various aspects of the findings have been discussed elsewhere: the implications of CASE tools for IS personnel (Orlikowski, 1988b); the changes in social relations and locus of power on project teams associated with use of CASE tools (Orlikowski, 1989); the change in organizational control mechanisms associated with use of CASE tools (Orlikowski, 1991); and the structuring nature of interactions among CASE tools, systems development work, key players, and the organizational context (Orlikowski, 1992).

**Table 1. Type and Amount of Interviews Conducted at SCC**

Position*	Applications Development	CASE Tools Support	Total
Partner (Senior Manager)	5	5	10
Manager (Project Manager)	27	15	42
Senior Consultant (Senior Analyst)	13	18	31
Staff Consultant (Junior Analyst)	11	4	15
Client (User)	11	10	21
Total	67	52	119

\* For comparison, job titles in parentheses provide the equivalent description used in traditional IS departments such as PCC.

**Table 2. Type and Amount of Interviews Conducted at PCC**

Position	Applications Development	CASE Tools Support	Total
Senior Manager	8	4	12
Project Manager	5	2	7
Senior Analyst	11	4	15
Junior Analyst	4	-	4
User	2	-	2
Total	30	10	40

the iterative approach of data collection, coding, and analysis was more open-ended and generative than in PCC, which focused on the development of concepts, properties, and relations, and on following the descriptions of how to generate grounded theory set out by Glaser and Strauss (1967) and Eisenhardt (1989). Because the study of SCC was part of a larger research project, the detailed write-up of the site and all the data generated by interviews, observations, and documentation were examined and coded by focusing on the change experiences associated with the CASE tools. This technique uses a form of content analysis where the data are read and categorized into concepts that are suggested by the data rather than imposed from outside (Agar, 1980). This is known as *open coding* (Strauss and Corbin, 1990), and it relies on an analytic technique for identifying possible categories and their properties and dimensions. Once all the data

were examined, the concepts were organized by recurring theme. These themes became prime candidates for a set of stable and common categories, which linked a number of associated concepts. This is known as *axial coding* (Strauss and Corbin, 1990), and it relies on a synthetic technique of making connections between sub-categories to construct a more comprehensive scheme. The SCC data were then re-examined and re-coded using this proposed scheme, the goal being to determine the set of categories and concepts that covered as much of the data as possible. This iterative examination yielded a set of broad categories and associated concepts that described the salient conditions, events, experiences, and consequences associated with the adoption and use of CASE tools in SCC.

These initial concepts guided the second field study conducted in PCC, allowing the process of

data collection, coding, and analysis to be more targeted. Following the *constant comparative analysis* method (Glaser and Strauss, 1967), PCC's experiences were systematically compared and contrasted with those of SCC. This analysis also used Miles and Huberman's (1984) technique for across-site pattern comparison and clustering, which involves matrix displays to compare key events, triggers, and outcomes (as evident, for example, in the contrasts of Table 3).

Data from PCC were first sorted into the initial concepts generated by SCC's data. It soon became clear, however, that the initial concepts generated by the first site did not accommodate some of the findings emerging from the second site. Accommodating PCC's experiences led to some important elaborations and clarifications in the emerging theoretical framework and forced a reconsideration of some of SCC's experiences. For example, the category "consequences of adopting and using CASE tools" did not include a concept having to do with client reactions to the CASE tools because this was not salient at SCC. PCC's experiences, however, indicated that client relations were indeed very relevant in shaping the interpretations and use of CASE tools and substantially influenced their effectiveness in the organization. Redefining the initial concepts to incorporate considerations of PCC's experiences required returning to the SCC data and re-sorting and re-analyzing them to take account of the richer concepts and more complex relations now constituting the framework. This ability to incorporate unique insights during the course of the study is one of the benefits of a grounded theory research approach, an example of what Eisenhardt (1989) labels "controlled opportunism," where "researchers take advantage of the uniqueness of a specific case and the emergence of new themes to improve resultant theory" (p. 539).

The iteration between data and concepts ended when enough categories and associated concepts had been defined to explain what had been observed at both sites and when no additional data were being collected at PCC or found at SCC to develop or add to the set of concepts and categories, a situation Glaser and Strauss (1967) refer to as "theoretical saturation." The resultant framework is empirically valid because it can account for the unique data of each site and can generalize patterns across the sites (Eisenhardt,

1989). The categories and concepts thus developed are shown in Table 3.

Precautions were taken to corroborate the interpretations made (Miles and Huberman, 1984; Yin, 1989a). Emerging concepts were checked for representativeness by examining them across participants and with multiple methods. For example, participants' reports of the criteria they used to evaluate which CASE technology to implement were checked against the documentation available from feasibility studies, while the purported youthfulness of PCC's IS department was checked by referring to personnel records. Triangulation across data sources (multiple informants at different levels of the firm, from different functional affiliations, and across sites) and across data collection methods (interviews, documentation, and observation), further served to strengthen the emerging concepts. The constant comparative method also requires the searching out and checking of contrasts and negative evidence, thus forcing the confrontation of emerging explanations with possible alternative ones. Finally, the participants in the study (particularly those at PCC) provided commentary, correction, and elaboration on drafts of the findings and framework.

## Research Results

The process of organizational change around CASE tools, developed from the two organizations' experiences, is depicted in Figure 1. The figure shows the categories and concepts that emerged as salient from the data analysis, as well as how they interact with each other. This process is proposed as an initial formulation of the key concepts and interactions that portray CASE tools as a process of organizational change. No claim is made that the concepts and interactions presented here are exhaustive. Further organizational change studies around CASE tools should add to or modify the ideas presented here—this is how we build on each others' work.

The organizational change process is influenced by the structural premise that human action and institutional contexts interact over time (Orlikowski and Robey, 1991). Starting on the left side of Figure 1, we can walk through this process. Initially, IS managers, influenced by their environmental, organizational, and IS contexts

Table 3. Organizational Change Around CASE Tools: Concepts and Findings

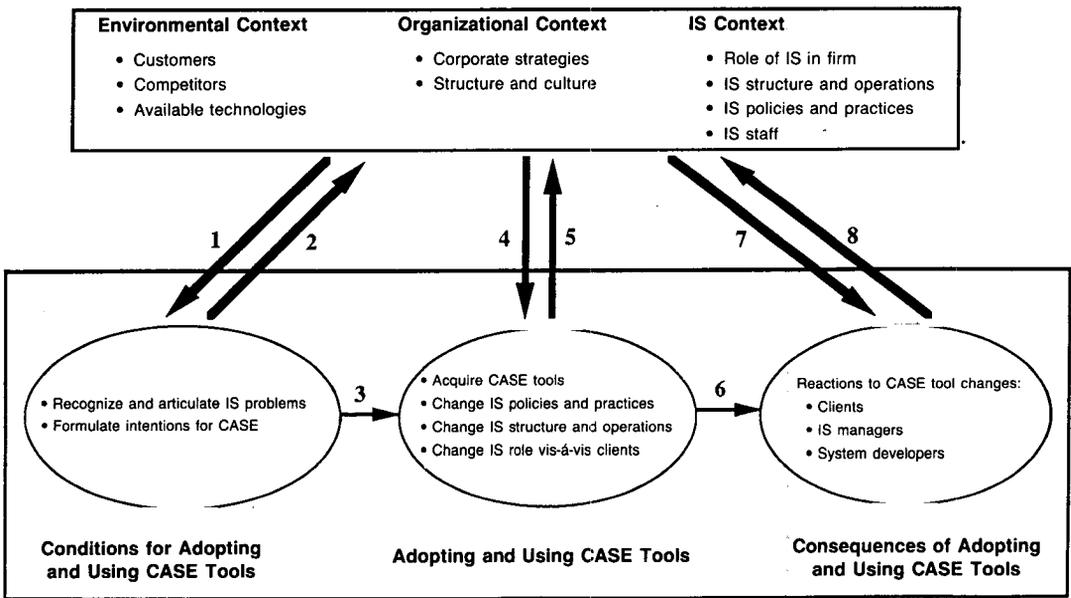
Categories	Concepts	Data (from SCC)	Data (from PCC)
<b>Environmental Context</b>	Customers	<ul style="list-style-type: none"> <li>Fortune 500 companies commission SCC to custom build typically large, transaction-oriented applications software</li> </ul>	<ul style="list-style-type: none"> <li>External customers purchase petro-chemical products; they care about speed, cost, and quality with which these are produced</li> </ul>
	Competitors	<ul style="list-style-type: none"> <li>Large software development houses building complex systems</li> <li>Some even providing hardware and maintenance support</li> </ul>	<ul style="list-style-type: none"> <li>Other major oil companies investing heavily in information technology for competitive advantage</li> </ul>
	Technologies	<ul style="list-style-type: none"> <li>Increasing number of hardware/software environments</li> <li>CASE tools recently appeared on the market</li> </ul>	<ul style="list-style-type: none"> <li>Increasing number of hardware/software environments</li> <li>CASE tools recently appeared on the market</li> </ul>
<b>Organizational Context</b>	Corporate Strategies	<ul style="list-style-type: none"> <li>Increase profitability for firm and partners</li> <li>Decrease time and costs of software development</li> <li>Leverage technical and managerial skills on project teams</li> </ul>	<ul style="list-style-type: none"> <li>Grow market share in petro-chemical business</li> <li>Decrease costs and streamline operations</li> <li>Use information technology for competitive advantage</li> </ul>
	Structure and Culture of Firm	<ul style="list-style-type: none"> <li>Matrix structure: controlled via hierarchy; teams operate from local offices</li> <li>Competitive culture reinforced by "up or out" career path: from staff consultant, through senior and manager, to partner</li> </ul>	<ul style="list-style-type: none"> <li>Divisional form, with business units having significant autonomy</li> <li>Hierarchical operating structure; entry level into firm typically through IS</li> <li>PCC promotes "career development" and "lifetime employment"</li> </ul>
<b>IS Context</b>	Role of IS in Firm	<ul style="list-style-type: none"> <li>Software development is SCC's core business since 1960s</li> <li>Application systems developed on a project-by-project basis for external clients from diverse industries in a custom-built, stand-alone mode</li> </ul>	<ul style="list-style-type: none"> <li>IS division recently centralized after being dispersed among the various separate business units</li> <li>Stand-alone applications commissioned/funded by business units</li> </ul>
	IS Structure and Operations	<ul style="list-style-type: none"> <li>Ad hoc project teams formed to work on client engagements</li> <li>Projects conducted at client sites, range from a few months to years, and include tens to hundreds of developers</li> <li>Project teams managed by partner and senior manager</li> </ul>	<ul style="list-style-type: none"> <li>IS division organized to mirror the structure of the company: one application group for each business unit</li> <li>Stand-alone systems for business units developed and maintained by application group dedicated to supporting that business unit</li> </ul>
	IS Policies and Practices	<ul style="list-style-type: none"> <li>In-house, standardized firm methodology, mandated on all projects, and in use for decades</li> <li>All new recruits trained in SCC's development methodology</li> </ul>	<ul style="list-style-type: none"> <li>No standardized firm methodology in place; each team adopts own approach</li> <li>Need to support the "200 plus" technologies makes a uniform set of policies and practices difficult</li> </ul>
	IS Staff	<ul style="list-style-type: none"> <li>13,000 systems developers at all levels throughout firm</li> <li>Recruits hired at entry-level from college—typically liberal arts, sciences, business, and engineering</li> </ul>	<ul style="list-style-type: none"> <li>320 full-time staff at all levels in the IS division</li> <li>All recruiting done from college at entry level, and employees can choose among application development, technical specialist, or line career paths</li> </ul>

<b>Conditions for Adopting and Using CASE Tools</b>	Articulating IS Problems	<ul style="list-style-type: none"> <li>• Increased competition from other software development firms</li> <li>• Increased demand from clients for larger, integrated applications</li> <li>• Increased risk as firm depends on specialized technical knowledge of system developers who have high turnover</li> </ul>	<ul style="list-style-type: none"> <li>• Lack of system integration; incompatible technologies in firm</li> <li>• Data quality problems: massive redundancy and inconsistency</li> <li>• Increased recognition that the information systems "grid-lock" and "mess" are hurting PCC's competitive position</li> </ul>
	Formulating CASE Intentions	<ul style="list-style-type: none"> <li>• Increased productivity</li> <li>• Decreased time and costs of systems development</li> <li>• Increased leverage of technical and managerial skills</li> </ul>	<ul style="list-style-type: none"> <li>• Competitive advantage</li> <li>• Development of an information architecture with shared data and shared applications</li> </ul>
<b>Adopting and Using CASE Tools</b>	Acquiring CASE tools	<ul style="list-style-type: none"> <li>• In-house development of integrated CASE tools based on existing firm practices and methodology</li> <li>• Dissemination of CASE tools to all offices in the firm</li> </ul>	<ul style="list-style-type: none"> <li>• Committee recommends strategic plan and development of integrated information architecture</li> <li>• IEM obtained, and then IEF CASE tools to enforce use of IEM</li> </ul>
	Changing IS Policies and Practices	<ul style="list-style-type: none"> <li>• Use of CASE tools mandated on all projects</li> <li>• CASE tools taught to all system developers</li> <li>• Adjustments to systems development policies and practices to accommodate the CASE tools</li> </ul>	<ul style="list-style-type: none"> <li>• Use of CASE tools mandated on all new development work</li> <li>• CASE tools taught to all system developers</li> <li>• New systems development policies (e.g., standards for shared data) and new development practices (e.g., cross-functional teams) established</li> </ul>
	Changing the IS Structure and Operations	<ul style="list-style-type: none"> <li>• No significant change in the organization of work, except that tool support specialists are included in projects</li> <li>• CASE tool skills augment existing systems development skills</li> <li>• System developers rely heavily on CASE tools</li> </ul>	<ul style="list-style-type: none"> <li>• New groups formed in IS division to build shared applications, support tools, and maintain shared data resource</li> <li>• Existing systems development skills replaced by IEM and IEF</li> <li>• System developers rely heavily on CASE tools</li> </ul>
	Changing the IS Role vis-à-vis Clients	<ul style="list-style-type: none"> <li>• Products delivered to clients remain custom-built, stand-alone systems, built and funded on a project-by-project basis</li> <li>• Incremental changes to systems delivered to clients are evident in the size and complexity of applications developed</li> </ul>	<ul style="list-style-type: none"> <li>• Applications no longer commissioned and owned by business units; applications and databases are built according to the IE strategic plan and "logical build sequence"</li> <li>• "Data ownership" is eliminated; data is accessible to everyone</li> </ul>
<b>Consequences of Adopting and Using CASE Tools</b>	System Developers' Reactions	<ul style="list-style-type: none"> <li>• Technically oriented system developers resent CASE tools, fearing loss of status, expertise, and marketability</li> <li>• Business-oriented system developers welcome CASE tools, favoring removal of tedious work and ability to spend more time on analysis</li> </ul>	<ul style="list-style-type: none"> <li>• Most system developers welcome CASE tools, influenced by their frustration with "systems mess," limited systems development experience, low investment in systems development status quo, and the belief that CASE will enhance skills, jobs, status, and marketability</li> </ul>
	IS Managers' Reactions	<ul style="list-style-type: none"> <li>• Perceive improved productivity and profitability</li> <li>• Believe CASE tools have leveraged technical and managerial skills</li> </ul>	<ul style="list-style-type: none"> <li>• Frustrated over reaction of business units</li> <li>• Attempt to deal with resistance and inertia; inability to get middle and lower levels of business committed to IEM, IEF, and architecture</li> </ul>
	Clients' Reactions	<ul style="list-style-type: none"> <li>• Minor (although they bear its cost)</li> <li>• Some are impressed by the technological sophistication of tools</li> </ul>	<ul style="list-style-type: none"> <li>• Resistance to redefinition of IS role and new policies around data sharing and sequence in which systems can be built</li> <li>• Conflict over apparent loss of power with respect to IS</li> </ul>

(arrow 1), recognize and articulate some problem with the systems development process and/or the kind of systems produced. In response to this interpretation, the managers choose to invest in CASE tools as a way of dealing with the perceived problem, formulating certain intentions for what changes they hope the CASE tools will occasion. In drawing on their contexts to articulate these interpretations and intentions, the managers also, and typically unintentionally, reinforce those contexts (arrow 2). The managers' actions—articulating a problem and formulating a solution—(labeled *conditions for adopting and using CASE tools*) lead managers to acquire a particular set of CASE tools and put them to use in systems development (arrow 3). In addition to being influenced by articulated interpretations and intentions, the IS managers' actions are influenced by the institutional context within which the tools are acquired and deployed (arrow 4)

(e.g., external context, such as available products on the market at the time, and internal context, such as corporate strategies). Having acquired CASE tools, IS managers may take action (labeled *adopting and using CASE tools*) such as changing IS policies, practices, operations, and relations with clients. In doing so, they affect the institutional context, changing it to a greater or lesser extent (arrow 5). Actions regarding adoption and use, in turn, result in various experiences and outcomes (arrow 6) as the key players—IS managers, system developers, and clients—act on and react to the changes (labeled *consequences of adopting and using CASE tools*). These actions do not occur in a vacuum but are influenced by the institutional context in which they occur (arrow 7). Similarly, action taken here by the managers, developers, and users will influence the institutional context (arrow 8), either reinforcing or changing it (Orlikowski and Robey,

**INSTITUTIONAL CONTEXT FOR ADOPTING AND USING CASE TOOLS**



**STRATEGIC CONDUCT IN ADOPTING AND USING CASE TOOLS**

**Legend for arrows:**

- |  |  |
|--|--|
| <p>1: Influence of context on IS managers' actions as condition for adoption of CASE tools</p> <p>2: Reinforcement or change of context as IS managers' actions are influenced by it</p> <p>3: Conditions that lead IS managers to acquire CASE tools</p> <p>4: Influence of context on acquisition and assimilation of CASE tools</p> | <p>5: Reinforcement or change of context as the CASE tools are assimilated</p> <p>6: Acquisition and assimilation of CASE tools leads to certain experiences</p> <p>7: Influence of context on reactions of key players around use of CASE tools</p> <p>8: Reinforcement or change of context as players are influenced by the context</p> |
|--|--|

**Figure 1. Process of Organizational Change Around CASE Tools (Key Concepts and Interactions Generated by SCC and PCC)**

1991). In the following section, these concepts and their interaction over time are discussed in detail for each of the two field sites.<sup>6</sup>

## SCC

### *Institutional context*

As depicted in Figure 1, the categories of environmental, organizational, and IS context were found to be relevant in influencing the adoption and use of CASE tools in SCC. The concepts constituting these categories are discussed in turn.

#### **Customers**

Because SCC is in the software business, IS services constitute the core business. SCC's customers, typically large Fortune 500 firms, purchase custom-built applications systems. Each SCC systems development project is a stand-alone engagement, with the system developers building an information system to meet specific functional requirements specified by the paying client. All work conducted on a client engagement is charged to the client, so the effort is concentrated on the specific system to be built. Occasionally, SCC obtains a follow-on contract with a satisfied client, which may be related to the previous system built.

#### **Competitors**

SCC competes with large software development houses as well as in-house development shops. It attempts to differentiate itself by building large, complex systems and has earned a reputation for successful systems development in various industries. While facing some competition, SCC is firmly established as one of the leaders in the software development market.

#### **Technologies**

The information technologies of the 1980s were represented by a plethora of hardware and soft-

ware environments. SCC has been forced to keep up to date with these new technologies because it has to work with them at client sites. It was also during the early 1980s when the notion of CASE tools began to gain credibility, and interest in using such tools to assist systems development grew quickly within SCC.

#### **Corporate Strategies**

SCC is committed to increasing revenues and profitability, which directly benefits the senior managers (owners) of the firm. In addition, there is interest in decreasing the costs of the core production process—systems development—as well as leveraging the existing technical and managerial skills in the firm.

#### **Structure and Culture**

SCC is organized as a matrix, with control of the systems development practice exercised hierarchically from the firm's headquarters, while the development work is conducted via ad hoc project teams operating out of local offices. The firm has a particularly competitive culture, which is reinforced by the strict, single career path that every employee follows. Employees advance in lock-step pattern from staff consultant through senior consultant and manager to senior manager over a period of about 10 years. The "up or out" nature of this career path ensures a high turnover within SCC, at a rate purported to be higher than normal for the IS industry.

#### **Role of IS**

SCC has been developing computer-based information systems for its clients across various industries since the 1970s. Although SCC also does some strategic planning and organizational development consulting, software development is its primary product and service. SCC is usually contracted to design and develop specific applications, with the installation, support, and maintenance of these applications being left to the clients. The applications are typically large, stand-alone transaction processing systems used to support major administrative activities.

#### **IS Structure and Operations**

SCC's production process is organized by project, with most of the systems development work

<sup>6</sup> Because the constant comparative method of analysis was adopted here, all concepts will be discussed for both organizations. Even if a concept emerged as relevant in only one site, it is nevertheless useful to consider its role in the other site so as to contrast the experiences of the organizations.

being conducted at clients' sites. Ad hoc project teams, varying from around 10 to over 100 personnel, are formed at the beginning of each client engagement. The projects extend from a few months to a number of years in duration and cost from a hundred thousand to a few million dollars. Each project is headed by a senior manager who oversees its general pace and direction and is administered by a project manager who exercises day-to-day control over project activities.

### IS Policies and Practices

For over a decade, SCC's work practices have relied on the firm's own standardized systems development methodology, which combines the techniques of Yourdon structured systems design, Warnier-Orr program design, and Chen's entity-relationship data modeling. SCC's policy is that all employees should be trained in the methodology and that all systems development work should be based on it. The methodology is thus well-entrenched within the company, in use on all SCC projects, and thoroughly documented in many large paper-bound volumes, as well as in computer form. Given the firm's practice of only hiring at entry level, this methodology comprises the primary production knowledge of SCC system developers. Senior managers attribute much of their company's success to the consistent use of the methodology across the firm. For example, a brochure published by SCC cites a senior manager as saying that the firm's methodology "reflects over 30 years of experience by SCC in developing and maintaining information systems in virtually every industry throughout the world . . . [it] is truly a proven approach."

### IS Staff

Because systems development is SCC's core business, it pays careful attention to how it hires and trains its system developers. It hires almost exclusively at entry level by recruiting at major colleges around the country. Most recruits have liberal arts, business, and engineering backgrounds, while a few have specialized in computer science. SCC spends considerable time and money training its recruits how to develop application systems using SCC's own systems development methodology. At the time of the study, there were approximately 13,000 system

developers employed at various levels throughout SCC.

## Conditions for adopting and using CASE tools

Two types of managerial action created the conditions that led to the adoption and use of CASE tools: (1) recognizing and articulating IS problems and (2) formulating intentions for CASE tools.

### Recognizing and Articulating IS Problems

During the 1980s, SCC's senior managers began to perceive increased competition from a number of other large software developers, as well as a growing demand from prospective clients for more integrated application software. They interpreted these pressures as threats to their competitive position and their profitability ratio.

### Formulating CASE Intentions

In response to this perceived threat, SCC's senior managers focused on increasing productivity and decreasing the time it took to build application systems. They reasoned that this could be accomplished through greater managerial leverage (that is, by increasing the number of consultants per senior manager and expanding span of control) and reduced dependence on technical skills. In the past, SCC had to ensure that system developers knew a range of programming languages, database management systems, teleprocessing software, and operating systems so they would be sufficiently versatile to operate in the multiple computer environments operated by their clients. Teaching the developers this knowledge took time and effort and required regular updating because such skills were technical and specific, quickly becoming obsolete as new computer products appeared on the market. SCC thus saw CASE tools as a way of decreasing costs, improving leverage, and increasing productivity. A senior manager noted:

The cost of building information systems is getting out of hand now. So we have to look at people cost, which is the most important component. SCC has to be tool-oriented given the sorts of fees we charge and the kinds of people we hire. We need to increase our productivity, as there is a scarcity of DP people, and people turn around a lot. . . . So SCC takes a high risk by

investing knowledge in individuals who may leave soon. We need to leverage that risk. We have to be able to divorce knowledge from programmers. That's why we invest in tools.

Likewise, another manager explained their intentions this way:

The skills issue is another motivator for us [to] use tools. It is expensive to educate people, particularly in the technical details. If we can embody the knowledge of technical experts in the tools, then a less experienced person can work with the tools and become an expert. So we factor out the complexities of systems development and embed these in the tools. Then we can forget about them because the tools insulate us from the technical environment.

### *Adopting and using CASE tools*

Following the decision to invest in CASE tools, managers introduced these tools into SCC by taking four types of action: (1) acquiring CASE tools; (2) changing IS policies and practices; (3) changing the IS structure and operations; and (4) changing the IS role vis-à-vis clients.

#### **Acquiring CASE Tools**

In response to their understanding of the key business issues facing the firm and their strategy for addressing them, SCC's senior managers initiated the adoption of CASE tools and disseminated their use throughout the firm. Instead of purchasing a set of CASE tools developed by an outside vendor, SCC managers decided to build their own tools because they had the requisite resources in-house to undertake such a task, and such a strategy would allow SCC to capitalize on the established expertise and experience with its systems development methodology.

Because SCC had a heavy investment in its existing methodology—with all the firm's work practices and knowledge predicated on it—SCC senior managers were concerned not to lose this asset as they began to use CASE tools. Consequently, they commissioned SCC's in-house technical support group to develop a set of integrated CASE tools that were based on the firm's standardized methodology and years of accumulated practice knowledge.

#### **Changing IS Policies and Practices**

Once the CASE tools were built, SCC senior managers established a new policy that mandated their use on all large projects. As one manager pointed out, the firm policy quickly became "tools, tools, and tools." Complementing this policy was the requirement that all system developers be trained on the tools. This was fulfilled either as part of the developers' annual continuing education courses or—where system developers needed to use the tools sooner—on the job at client sites, with technical specialists conducting ad hoc CASE training classes. Despite the new policy, the work practices of SCC developers were not significantly changed by the implementation of CASE tools because these tools had been deliberately constructed to support SCC's established systems development methodology and existing work practices. A senior systems analyst at SCC 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.

Thus, beyond being mediated by the CASE tools, systems development in SCC was not changed significantly through the adoption of the CASE tools. Indeed, the tools, because they reflected SCC's existing systems development methodology, reinforced the firm's established work practices and norms. As one manager noted, "The methodology has not been affected by the tools, as we are not fundamentally changing the direction or discipline of our work."

#### **Changing the IS Structure and Operations**

No major structural changes to accommodate the CASE tools were deemed necessary, and none were implemented, although some operational adjustments were made on the projects. Procedures and checklists were incorporated into project schedules to ensure that the tools were applied correctly, while technical specialists were included on project teams to ensure that the CASE tools were available, reliable, and suitably

customized to clients' environments. Responsibilities around systems development had shifted somewhat, with technical and supervisory skills leveraged when the tools facilitated the abstraction of systems development knowledge from the underlying hardware and software environment. This served to monitor and coordinate systems development work. The system developers began to spend more time with users doing business analysis and more time in front of workstations interacting with the tools. For many developers in SCC, particularly those who had had no other experience with systems development, this reduced their need to acquire technical knowledge while it increased their dependence on the CASE tools. While knowledge of and experience with the CASE tools became a prerequisite for advancement in the firm, no new career paths were introduced.

### Changing the IS Role vis-à-vis Clients

After the adoption of CASE tools, there was little attempt on the part of senior managers to change SCC's interactions with its clients. Jobs were still sold on a project-by-project basis; systems that were built were still largely stand-alone, and clients still paid for all the systems development work performed by SCC employees. What did change, however, was that the CASE tools became part of the price of the project. Thus, added to the total project cost were the cost of installing hardware at the client site to run the tools, the cost of full-time technical specialists to support the tools, and the cost of training developers and client-users participating in the project on the use of the tools. Client relations, thus, remained essentially the same, and the use of CASE tools served to reaffirm rather than redefine SCC's role with respect to its clients.

### Consequences of adopting and using CASE tools

Various consequences followed SCC's adoption and use of CASE tools, and these are expressed in terms of the reactions of three key players: clients, IS managers, and system developers.

#### Client Reactions to CASE Changes

Because there appeared to be little change in client relations following the dissemination of

CASE tools within SCC, it is not surprising that there was no marked client reaction to the changes. SCC system developers continued to deliver custom-built, stand-alone information systems to the clients, just as they had always done.

The nature of the application systems being delivered to clients was largely unaffected by the tools, although a few project managers noted that the applications being built with CASE tools tended to be larger or more complex than would have been feasible had CASE tools not been used. It seems that knowing that CASE tools would be available during systems development raised expectations about the possibility of building larger systems with more features and greater functionality. Beyond these product enhancements, however, the CASE tools did not significantly influence client experiences and actions.

#### IS Manager Reactions to CASE Changes

SCC managers believed that use of the tools had led to greater productivity (although this was based more on intuition than any formal measurement) and an increased ability to leverage senior and technical personnel on projects. One senior manager said:

With tools we can make money leveraging people, that is, having many bodies even at lower rates can be profitable with tools. . . . [The] tools allow us to leverage inexperienced people on our project. So we can take a kid out of school, let's say with a major in English, and in a very short time he can achieve high productivity, that is, achieve the productivity level of a client programmer with 10 years' experience. It's not that we are making him more effective at his job, but we are dumping him into a production environment that has 10 years of programming experience built into it.

From the viewpoint of managers, tools decreased the firm or project's dependence on specific individuals, whose particular knowledge of specific software and hardware configurations were difficult if not impossible to replace. With tools embodying complex technical expertise as well as knowledge of specific application systems, system developers became more replaceable and interchangeable. As a result, the risks and costs of large-scale systems development were seen to have been reduced.

### System Developer Reactions to CASE Changes

The reaction of SCC system developers to the changes accompanying the CASE tools centered on individual issues of skills, work experiences, and learning opportunities. They were less concerned with the impacts of CASE tools on the overall productivity of the firm, or the effectiveness of particular projects. Rather, they were interested in realizing their own individual aspirations, both in the short term (e.g., acquiring particular knowledge or specific experience), and the long term (e.g., attaining career goals, either within SCC or in the larger business community). Two different reactions to the CASE tools were evident at SCC, one rooted in a technical and the other in a business orientation toward systems development.

**Technical Orientation.** Many of the system developers who planned a career in IS (e.g., as system designers, analysts, or managers) perceived CASE tools as a threat to their hard-earned skills (e.g., knowledge of operating systems and programming languages) and experiences with specific hardware and software systems. Concerned with job security and career mobility, these developers believed that their marketability was depreciated by the proliferation of CASE tools. They feared that demand for technical expertise (traditionally high for IS personnel) would decline as a result. Some of these system developers also felt less personally involved in their work, mediated as it now was through the tools. One junior analyst resigned in protest over having to use tools, noting, "I just couldn't take this generated code stuff, and not having any input into the programming. . . . Also the things I was learning are not marketable."

Having automated users' tasks for years, system developers were beginning to experience the abstracting influences of automation in their own work. A senior analyst observed that CASE tools limited the possibility of acquiring and using technical knowledge in systems development because using tools amounted to a "form-filling exercise." He felt this had long-term implications for those developers who "don't know the underlying reasons for things. They don't know why they do things in a certain way. So we find that they are almost helpless without the tool

environment. They are only able to find canned solutions to standard problems."

There was also a perception among many of the technically oriented developers that tools had decreased their opportunities to learn new technologies. On one of the projects, a new database management system was introduced, and system developers eagerly anticipated learning a new, sophisticated product that they perceived would be intellectually challenging and would enhance their skills portfolio. Their excitement was short-lived, however, for they soon realized that the database system would only be accessible by the CASE tools and that they would have no direct contact with the new product. A sense of lost opportunity was sharply felt and resented.

**Business Orientation.** Most of the system developers whose career goals were oriented toward business or general management had a different reaction to CASE tools. These individuals did not perceive their careers in terms of systems design or IS management and did not see tools as threatening their skills or marketability. For such system developers, the tools provided welcome relief from the tedium and complexity associated with the technical details of large application systems.

These developers believed the CASE tools expedited their work, allowing them to spend more time on the tasks they enjoyed (such as business analysis), and that the tools helped them appear more productive and hence more valuable to their projects and SCC. One analyst noted, "Tools do prevent us from learning technical details. And that's good, as no one comes to SCC to be a programmer." And a senior analyst observed, "I don't want to turn out delightful code or make the coding activity creative. I must get the job done simply, and tools make things simple." Because these developers attached little value to the technical expertise now embedded in CASE tools, they perceived the tools as facilitating their work rather than as jeopardizing their expertise or status.

A discussion of the general patterns and implications of SCC's CASE experiences is taken up in the discussion section.

## PCC

### *Institutional Context*

As with SCC, the environmental, organizational, and IS contexts of PCC provided important information about its experiences with CASE tools. Because PCC's IS division is internal, its environmental context includes aspects of the environment outside PCC, as well as that inside PCC but external to the IS division.

### **Customers**

PCC customers purchase petroleum and chemical products, unlike the software products purchased by SCC customers. While information technology is not a critical component in the products purchased by PCC customers, it does influence the speed, cost, and quality with which products are produced and delivered (e.g., electronic billing at gas service stations). The IS division's customers are other business units within PCC, which use information services to meet their units' particular functional requirements.

### **Competitors**

Major oil companies make up the bulk of PCC's competitors, and PCC's senior managers believe these competitors are investing heavily in information technology across both upstream and downstream value chains. This has put pressure on the IS division to provide more competitive technology to the various business units. As a senior manager commented, "Information technology is one of the few levers that we must pull to gain competitive advantage."

### **Technologies**

Like SCC, PCC faces an increasingly diverse array of hardware and software environments, and given the range of its operations (e.g., oil pipelines, chemical processing, inventory and customer support), the IS division has to keep on top of many developments in various technological areas. The development and marketing of CASE tools in the 1980s generated interest among some IS managers, who wondered if these tools could help PCC's systems development challenges.

### **Corporate Strategies**

At the time of the study, PCC was interested in both growing its existing business (increasing market share) and streamlining its operations by cutting costs and redesigning business processes. There was a heightened recognition among senior managers that information systems were critical to running their business and that these should be more effective. Using information technology to gain competitive advantage thus became a general corporate strategy.

### **Structure and Culture of Firm**

PCC is organized divisionally with business units representing the major areas of the business (e.g., research, refining, sales, human resources, etc.) The firm is mature and well-established within its industry. Its culture was described by one manager as "paternalistic," and this is evident in its conservatism and strong emphasis on quality of work life and career development programs. PCC has no mandatory "up or out" career policy. On the contrary, it is widely known, as one senior analyst commented, to "hire people for life."

### **Role of IS**

PCC's IS division was formed in the late 1980s, representing a centralization of resources from dispersed business units into a separate division under the vice president of technology, who reported to the company's president. Before consolidation, systems development work in PCC was handled by distributed IS groups located within separate business units. Information systems built under this arrangement were local in scope and funding and were used solely by individual business units. After consolidation, IS still developed specific information systems for the functional requirements of individual business units. These systems (and their data) were commissioned on an "as-needed" basis by the business units and then subsequently paid for and "owned" by these business units.

### **IS Structure and Operations**

While IS had been consolidated functionally, its structure and operations still reflected its distributed origins. The division was organized into separate application groups, whose

members represented and serviced the separate business units they had supported in the past. Indeed, some of the system developers even retained their physical location within the various business units "to remain close to the users." There was little coordination across the application groups, with each group essentially being funded by and focused on its own internal clients. There was thus little incentive to consider the corporate implications of application systems.

### IS Policies and Practices

While the IS function had been consolidated, its distributed history left a legacy of incompatible technologies and non-uniform IS practices and policies. These divergent approaches to systems development were continued after consolidation; each application group continued to service its specific clients as it had done before. No standardized systems development methodology or common set of standards and procedures was enacted. As a result of this eclecticism and the "200 plus technologies" deployed within PCC, system developers' knowledge and practices were broad and diverse, reflecting the traditional dispersion and decentralized focus at PCC.

### IS Staff

Like SCC, PCC also hired its system developers at entry level from college, although it typically drew from computer science or engineering programs. Within the IS division, employees could choose to follow one of three career paths: two in IS—application development and technical specialization—and one into the business out of the IS division. Indeed, as a senior IS manager noted, "We're the entry point into the rest of the firm. PCC believes that some technology training is valuable to everyone. So people spend three to five years here before moving into the professional side of business." At the time of the study, there was a total of 320 full-time members of the IS division. These included managers and individuals engaged in application development, maintenance, and technical support activities.

### *Conditions for adopting and using CASE tools*

Two types of action created the conditions that led to the adoption and use of CASE tools: (1)

recognizing and articulating IS problems and (2) formulating intentions for CASE tools.

### Recognizing and Articulating IS Problems

With the IS consolidation, the managers of the newly formed IS division assessed the role and status of information technology within PCC. They found what one senior analyst described as a "complex mosaic of bridges, data redundancy, and data integrity problems" and a set of information systems that a manager characterized as "spaghetti junction." Centralization had bestowed upon the IS division a wide diversity of hardware and software with little or no integration, and an extensive duplication of corporate data. The company was reputed to have as many as 20 product files and—as one technical specialist graphically put it, "as many customer databases as Kellogg's has corn flakes." Senior IS managers interpreted this chaotic state of information systems to be hurting PCC competitively. Three of them 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.

Data were dispersed throughout the company, coding schemes were inconsistent and different, and people were drawing incorrect conclusions from poor data.

We came to the realization that IS was not serving the business as it should.

### Formulating CASE Intentions

Given their articulation of the state of information systems at PCC, IS managers believed that to gain competitive advantage through information technology they would have to significantly improve the quality of PCC's data and integrate its disparate systems. Two IS managers explained:

Senior business people were dissatisfied with the systems. They were a bunch of spaghetti. They wanted a shared environment to get good information.

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. The value added now is cross-functional coordination.

Based on their assessment, senior IS managers commissioned an IS strategy committee to examine alternative ways to deal with the "systems gridlock." After six months of deliberations, the committee proposed a long-term IS strategy for PCC, which called for the construction of a corporate-wide information architecture. This proposal was premised on the belief that a shared corporate data environment, providing timely data to appropriate users, would generate changes in business practices and lead to improved firm performance. To realize this architecture, the committee recommended the adoption of a standard systems development methodology and CASE tools.

### *Adopting and using CASE tools*

Four types of actions were relevant in understanding the move toward CASE tools in PCC: (1) acquiring CASE tools; (2) changing IS policies and practices; (3) changing the IS structure and operations; and (4) changing the IS role vis-à-vis clients.

#### **Acquiring CASE Tools**

To implement the committee's broad vision, the IS division acquired the information engineering methodology (IEM), an approach to systems planning, analysis, and design based on shared data (Martin 1990a; 1990b; 1990c).<sup>7</sup> An IS manager explained the committee's conviction that the information architecture, methodology, and tools were all indispensable: "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, [or] else we'll retreat to our prior bad habits." The IS strategy committee had recommended that PCC adopt a set of integrated CASE tools that were compatible with the selected methodology. While two products on the market were compatible with IEM—information engineering workbench (IEW) from Knowledgeware and Texas Instrument's information engineering facility (IEF)—only the latter

was deemed sufficiently integrated for PCC's requirements. A few months after standardizing on IEM, PCC acquired the IEF CASE tools. Significantly, the IS managers stressed that their adoption of CASE tools was not targeted at increasing productivity. One manager commented, "We went to [CASE tools] to implement the architecture rather than for efficiency gains."

#### **Changing IS Policies and Practices**

The adoption of the IE methodology and IEF CASE tools created, for the first time within PCC, a single, standardized approach to systems development. IS managers mandated the use of the standardized approach on all new systems development projects. They arranged for extensive training programs for IS personnel involved in new systems development, and they hired consultants to help establish new IE-based systems development standards and procedures, as well as policies for sharing the data and applications comprising the corporate information architecture.

#### **Changing the IS Structure and Operations**

IS managers signaled their commitment to the new methodology and tools by making a number of structural and operational changes. The IS division was reorganized and three new groups were created: (1) an architecture group (to develop the corporate-wide architecture); (2) a shared applications group (to develop and maintain cross-functional applications); and (3) a data and technology group (to manage the firm's shared data resource and to support the CASE tools). The separate IS groups that had serviced individual business units were retained to maintain existing stand-alone systems and to assist local end-user computing.

Many systems development jobs had changed as a result of the structural changes in the IS division—some developers moved to the shared applications development group, while others joined the technical support group to maintain the new shared databases and CASE tools. Most of the IS employees experienced major changes in their responsibilities, skill sets, and work norms. For example, they had to learn the new concepts and techniques of IEM and IEF, as these had become the only sanctioned approach to systems development. Further, development of shared ap-

<sup>7</sup> At the corporate level, IEM 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, IEM provides a detailed project view of data and specific development techniques with which to design and construct specific systems within the infrastructure.

plications required considerable negotiation with users and members of other project teams to resolve issues around the interpretation, definition, and handling of shared data.

System developers and users were not accustomed to thinking of systems or data beyond the boundaries of separate functional units because such a practice previously had been unnecessary and inconceivable. Making this change at PCC required new training courses within the IS division that imparted skills of negotiation, relationship-building, and conflict resolution—competencies that, as one IS manager indicated, had never before been considered part of the narrow technical role played by IS. A senior IS manager commented on the changes experienced by his division: "There has been significant cultural change in the systems group due to tools."

### Changing the IS Role vis-à-vis Clients

PCC's IS managers hoped that the use of CASE tools—in conjunction with their implementation of the IE methodology and corporate architecture—would facilitate the redesign of business processes in the business units as well as in IS. This attempt to influence the business represented a significant departure from the reactive relationship that IS had traditionally maintained with its internal business clients. Instead of developing stand-alone functional systems for internal business units as commissioned, the IS division now took a proactive stand toward information systems within PCC.

IS designed an information architecture and identified and initiated a sequence of systems development projects to construct this architecture. By insisting on a corporate architecture—composed of shared databases and shared application systems—IS had redefined its role in the firm. Instead of responding to the business units' requests for systems, the IS division now dictated (on the basis of the strategic plan and information architecture) which systems would be built, when, and how, as well as what data would be collected, stored, and accessed, and by whom. It further insisted on cross-functional development projects and required common funding of shared databases and applications.

## Consequences of adopting and using CASE tools

The consequences of adopting and using CASE tools can be categorized in terms of the reactions of clients, IS managers, and system developers.

### Client Reactions to CASE Changes

Despite the fact that one of the documented objectives of adopting the IE methodology and CASE tools was to "transform the business and business practices," little business change had occurred. The business units had not implemented any modifications in structure, work practices, or norms as a result of the corporate-wide information architecture and IE business analyses completed by IS. Instead, PCC business managers reacted to the IS initiatives by ignoring or resisting 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.

The clients' opposition stemmed from their perception that the changes instituted by the IS division threatened established organizational norms, practices, and privileges. For example, the notion of a corporate information architecture with shared access to data ran counter to PCC's long-standing policy on information sharing, which restricted access to data to only those people with a clearly defined and authorized "need to know." The shared data policy inherent in the information architecture reversed this policy by allowing access to anyone who was interested. A senior business manager explained:

We used to have a closed information environment; that is, all doors are closed unless you prove otherwise. Now [with IE] 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 units to accept it has been a serious stumbling block in IS's attempt to change the kind of systems delivered. Two senior IS managers commented:

This sort of paradigm shift is difficult to implement as people are reluctant to force searchlights into dark corners. It's very political.

It's at the business unit levels where we have met the most resistance. Our reward systems are a problem. They don't reward sharing.

Reinforcing this resistance was the business units' focus on their own short-term needs and goals. An IS manager explained the impact of this on IS:

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.

### IS Manager Reactions to CASE Changes

Most IS managers were very frustrated with the resistance of the business to the development of the architecture and the attempt to make data accessible across the corporation. They endeavored to overcome the business units' resistance by outlawing the concept of data ownership. A manager explained that the word "ownership" had been dropped from their vocabulary, because "it is too possessive, too explosive, and too functional." While helpful, these proactive steps were not entirely successful because they too were bound up in expectations about the appropriate role of the IS division. While IS had redefined its role and received approval for it from PCC's senior management, such a shift had not been clearly communicated to or accepted by the business unit managers and their users. A senior technical specialist observed that the change in IS's role was counter-cultural and hence difficult for the business units to accept:

Here at PCC, 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.

An IS manager confirmed this perspective, noting:

Right now, IEF is driven by the systems community. ...All the [planning] was done by systems people—some with good business background—but not many business people were involved. It was probably driven too much by the systems community, which may be why there is so much consternation about the architecture.

Not having substantial political clout in the organization, the IS managers felt somewhat immobilized by the unexpected and hostile reaction of their clients. Some considered involving PCC's senior management in an attempt to get top-down enforcement of the IS plan. A project leader commented:

The nature of the bureaucracy here at PCC is that the customer comes first, and so we will slap in something quick and dirty. It will take us a long time to get an architected solution because you can't stop people doing this if they can rationalize it for the customer. . . . We've sold IEF and the shared data concept to senior business management, but we have not filtered this message down to the masses. It is just as important to get buy-in from business users as it is to get senior management commitment. I see this all the time. We say, "This is the way the corporate level wants to do it," and the user on our team says, "I'm the one using the system, and I'm still going to do it the way I want to." ...Senior management has to talk to their own people as we have to get real users involved and committed if this is ever going to work.

### Systems Developer Reactions to CASE Changes

In contrast to SCC, PCC developers' reactions to CASE tools did not differ by technical or business orientation. Most PCC developers welcomed the IE methodology and the CASE

tools, despite having experienced significant changes in work practices, norms, skills, and responsibilities. More than three quarters (77 percent) of the 26 system developers interviewed indicated that the introduction of tools was a positive move, both for PCC and for their individual careers. In attempting to understand this unexpected reaction, the researcher searched for clues in the data that might account for such enthusiasm and identified five potential explanations. The first three—frustration with the status quo, believing the CASE strategy, and valuing CASE skills—reflect developers' perceptions, interpretations, and experiences. The fourth—willingness to change—involves an individual attribute of the developers, and the fifth—human resource policies—is a property of the firm.<sup>8</sup> These five explanations are elaborated on below.

**Frustration With the Status Quo.** Many of the developers had experienced first-hand the problems of PCC's "systems mess" and believed that IEM and IEF could bring much-needed order to a chaotic situation. One project leader expressed his frustration this way: "We were trying to compete in the Indy 500 in 1989 with a 1962 motor car." Yet another noted, "Because of the lack of integration among our systems, when you built a new system you often had to build 10 or 12 different interfaces. So the pain, agony, and cost of building a new system in our environment, and to maintain it, were enormous."

**Believing the CASE Strategy.** Many of the developers believed that the strategy behind the standardized methodology and CASE tools—to build high-quality, integrated systems and to share data throughout the firm—would enhance PCC's competitive advantage. In contrast to the system developers at SCC, who perceived their firm's strategy around CASE tools as attempting to decrease costs, reduce dependence on developers' expertise, and improve senior managers' incomes, PCC's system developers believed that CASE tools would yield better systems for PCC. One senior analyst remarked, "I know [PCC] is trying to use CASE tools to build an architecture, and so we have to enforce rigor and consistency in systems development so we

can share data across the corporation. We see tools as a means to do this."

**Valuing CASE Skills.** PCC system developers believed that their new skills in business analysis and CASE tools were valuable and enhanced rather than diminished their future marketability. This perception was helped by the fact that the IEF CASE tools had high market penetration relative to other CASE tools, in marked contrast to the limited diffusion of SCC's in-house tools. A senior analyst commented:

IEF provides a core set of skills that are highly marketable. There will be a lot of demand for experienced users of IEF and people who know how to use them intelligently. ... PCC analysts see IEF as a new skill, on the leading edge, that makes them more marketable.

Many of the PCC system developers also believed that their work had improved as a result of CASE tools. One analyst indicated that his job was more pleasant now, noting, "The tool has automated the drudgery of my job." Another remarked:

I enjoy my job more now as I feel more like an analyst. ... In the past, new hires would be the grunt programmers, and a senior person would be the business analyst. Now new hires can deal with users, and this provides a better sense of accomplishment for people who are not technical hackers.

Viewing CASE tools as enhancing skills and expanding jobs was related to the perception by some of the junior developers that tools had augmented their status vis-à-vis their seniors. A junior analyst stated, "[The tools] are an equalizing force. Everyone is thrown back to zero. This is good for me just coming into the firm. I can come out looking better than people my senior."

**Willingness to Change.** The shared applications development group—the group in the front line of CASE tools use—was populated by system developers without long, technical IS careers to their name. Because the IS group hires at entry level, its developers had accumulated less experience and technical skills than comparably sized IS departments in other organizations. Two senior IS managers observed:

Most of our people are younger, so they don't have much invested in the old traditional ways of doing systems development.

<sup>8</sup> While treated separately here for analytical reasons, these explanations are clearly interdependent and may even overlap. An examination of such interactions and overlap cannot be attempted here, but the issue warrants further research.

As they're younger, they don't carry much baggage and so have accepted IEF/IEM quicker, easier, and more enthusiastically.

The manager of the shared applications development group echoed this view, suggesting that the experience of his group members critically influenced their attitude toward the tools: "Learning IEF is like learning a second language. A person with 15 years experience will have a harder time adapting to it than a younger person. ... I got lucky—my shop is younger, thirtyish, and my most senior project leader is only 27 years old."

To confirm this claim, the age distribution of the shared applications group was obtained. The average age in this group of 44 members (including managers) was 33 years (s.d. of 5.98; range of 20 to 46), with two-thirds of the group being less than 35 years old. That this somewhat youthful group was more willing to change is likely influenced by the fact that they have spent less time in systems development and hence have a lower investment in the status quo—in either their personal skill portfolios or traditional systems development practices—than would developers who have spent more years building up a set of skills and establishing cognitive and behavioral habits around systems development.<sup>9</sup>

**Human Resource Policies.** PCC had a well-respected set of human resource strategies that mitigated much potential anxiety around organizational change. Like SCC, PCC hires its system developers straight after graduation from college, but unlike SCC, PCC claims to hire people "for a career" and to treat them as "life-time employees." Under such conditions, PCC system developers—even those who may opt for the technical specialist career track—are less likely to interpret the tools as a threat to their job security or career advancement. Two analysts commented:

This company is progressive. They promote career development, encourage new things, and rotate you a lot, every two to three years.

PCC is excellent at promoting its people's self-development. There's a belief here that people

are a number one resource, so you need to invest money in your people. We have an extensive human resources system to check on your development. We have a lower turnover than the norm—2 percent for our systems group, 25 percent in other companies. We hire a lot of outside contractors to do grunt work. We have better working conditions as well, reasonably good salaries, and less working hours.

An IS manager concurred, noting that, "At PCC, we're committed to our people," observing that the firm has a policy not to lay off people and that the IS division makes a practice of hiring contractors to provide a float when more or fewer systems people are needed.

In the following discussion, the organizational changes experienced by PCC around CASE tools are contrasted with those of SCC. Table 3 includes a summary of both companies' experiences with CASE tools.

## Discussion

While SCC and PCC both implemented CASE tools within their systems development practices, their experiences differ significantly. The comparative analysis method of grounded theory, which allows contrasting SCC with PCC on a common set of concepts (see Table 3), suggests that these differences can be attributed to variations in the change process, the organizational context, and the intentions and actions of key players around the adoption and use of the CASE tools (see Figure 1). While these two sites have yielded a grounded theoretical framework of organizational changes associated with the adoption and use of CASE tools, it is possible to try and generalize the patterns discerned. By attempting to connect the grounded theory with aspects of existing formal theory, a more general substantive theory can result (Glaser and Strauss, 1967). Eisenhardt (1989) also advocates this approach, noting: "Overall, tying the emergent theory to existing literature enhances the internal validity, generalizability, and theoretical level of theory building from case study research" (p. 545).

A useful classification in the innovation literature is that of incremental and radical types of innovation (Dewar and Dutton, 1986; Ettl, et al., 1984; Pennings, 1988; Tushman and Romanelli, 1985).

<sup>9</sup> It is interesting to note that SCC also hires at the entry level, so willingness to change is not a function of experience alone but likely interacts with other factors, such as career orientation and human resource policies (see footnote 8).

These concepts can be applied here to characterize the different intentions and actions around CASE tools in the two organizations. *Incremental change* represents an extension of the status quo, that is, adjustments or refinements in current products, practices, relationships, skills, and norms. Such changes represent "minor improvements or simple adjustments in current technology" (Dewar and Dutton, 1986, p. 1423). They serve to tacitly reinforce present understandings as well as the established configuration of interests and interest groups. *Radical change* goes beyond augmenting the status quo, requiring a shift to fundamentally different products, practices, relationships, skills, and norms. It involves adopting a different paradigm, a step that typically disrupts the established pattern of understandings and interests.

To illustrate the difference between incremental and radical changes, consider the comment made by an IS manager at PCC describing the new IS world his group was attempting to usher in:

Let me give you an analog as 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 we need jigsaw pieces with flexible edges, which would allow us to change the way we do business.

An incremental change amounts to improving individual jigsaw pieces that nonetheless leave the existing jigsaw picture intact. A radical change requires reconfiguring the entire jigsaw and its pieces so that a totally different jigsaw picture emerges.

Using these concepts of incremental and radical change, we can characterize SCC and PCC's different experiences with CASE tools. Within SCC, the senior managers did not intend CASE tools to fundamentally change their practice of producing and delivering information systems to external clients. Indeed, they had specifically designed their CASE tools to embody existing practices and knowledge. They intended merely to improve

the productivity and decrease the costs of their existing way of doing systems development. As a result, no major changes to IS policies, practices, structures, or client relations were instituted, and the firm experienced increased efficiencies and leveraging of technical knowledge without radically transforming its systems development practice or the products delivered to clients. Thus, in terms of both the process and product of systems development, SCC intended and enacted incremental change with CASE tools. Within PCC, on the other hand, the senior managers intended to transform not only their own systems development practice but also the business processes and role of IS in the firm. They enacted a series of changes that fundamentally changed PCC's systems development process to reflect the IE methodology and IE-based CASE tools, and they transformed the IS product from stand-alone functional systems to a corporate-wide information architecture. PCC, thus, intended and had begun to use CASE tools to enact radical change in both the process and product of systems development.

These specific findings can now be represented more generally by expressing them in terms of two dimensions: (1) the *nature of change* (incremental and radical), and (2) the *locus of change* (process and product of systems development). These dimensions yield a four-fold classification of changes associated with CASE tools (depicted in Figure 2). Normann's (1971) labels for incremental change ("variations") and radical change ("reorientations") can be used to conveniently characterize each of the quadrants.

Such a classification, along with the process model of CASE changes presented in Figure 1, can be used—either *ex ante* or *ex post*—to explain, anticipate, or evaluate the organizational changes associated with CASE tools. Where IS managers introduce CASE tools to improve the existing process of systems development through increasing productivity or cutting costs, organizations will likely experience *process variations*. Where the CASE tools are used to improve the product delivered to clients, without significantly altering its nature, ownership, or delivery arrangements, organizations will likely experience *product variations*. The change process enacted (expressed in terms of context, initiating conditions, strategic conduct, and resultant conse-

quences) will likely resemble that experienced by SCC (represented in Figure 3).

On the other hand, where managers implement CASE tools to substantially change how systems are developed through radically changing the process of systems development, organizations will likely experience *process reorientations*. Where managers use CASE tools to significantly change the nature of the product delivered to clients, including ownership and delivery arrangements, organizations will likely experience *product reorientations*. The change process enacted will likely resemble that experienced by PCC (represented in Figure 4).

While Figures 2, 3, and 4 depict the content and process of organizational change that may be experienced by organizations adopting and using CASE tools in difference situations, it is important to keep in mind that the relationships depicted are not deterministic. The organizational changes occasioned by the adoption and use of CASE tools are not always those intended by key players. First, process and product variations/reorientations may be the result of unintended consequences. For example, key players may intend an incremental change, but due to unforeseen factors, such as more complex software or a poorly conceived implementation plan, these may become more fundamental than anyone an-

anticipated and be experienced as radical, not incremental, change. Second, unintended changes may occur that go well beyond the specific process and product variations/reorientations intended. For example, in conditions such as those at PCC, the tension between the IS division and the business units may, over time, become so unsatisfactory that users seek IS services outside of the organization. Alternatively, in conditions such as those at SCC, the system developers' technical knowledge may eventually become so depleted that they become significantly less productive in non-standard, non-tooled environments.

### Implications for Research and Practice

This paper has presented the findings of a grounded theory study into the adoption and use of CASE tools and has developed a theoretical framework for conceptualizing this as a process of organizational change. The process represented in Figure 1 and the framework of CASE intentions given in Figure 2 provide one way of examining these organizational changes around CASE tools. By taking into account the existing institutional context and IS managers' intentions for and actions around CASE tools, the nature

		LOCUS OF CHANGE	
		Process of Systems Development	Product of Systems Development
NATURE OF CHANGE	Incremental Change	Process Variation <i>[SCC]</i>	Product Variation <i>[SCC]</i>
	Radical Change	Process Reorientation <i>[PCC]</i>	Product Reorientation <i>[PCC]</i>

Figure 2. Nature and Locus of Change Associated With CASE Tools

and locus of changes associated with CASE tools (as well as the consequences of these changes) can be anticipated, explained, and evaluated. The framework developed here does not imply that CASE tools per se *cause* process and product variations (as in SCC's case) or process and product reorientations (as in PCC's case). Rather, as shown in Figures 3 and 4, it indicates that such changes emerge from particular interactions of institutional context, key players' intentions and actions, and the CASE technology. The findings and framework articulated here have implications for both the research and practice of systems development.

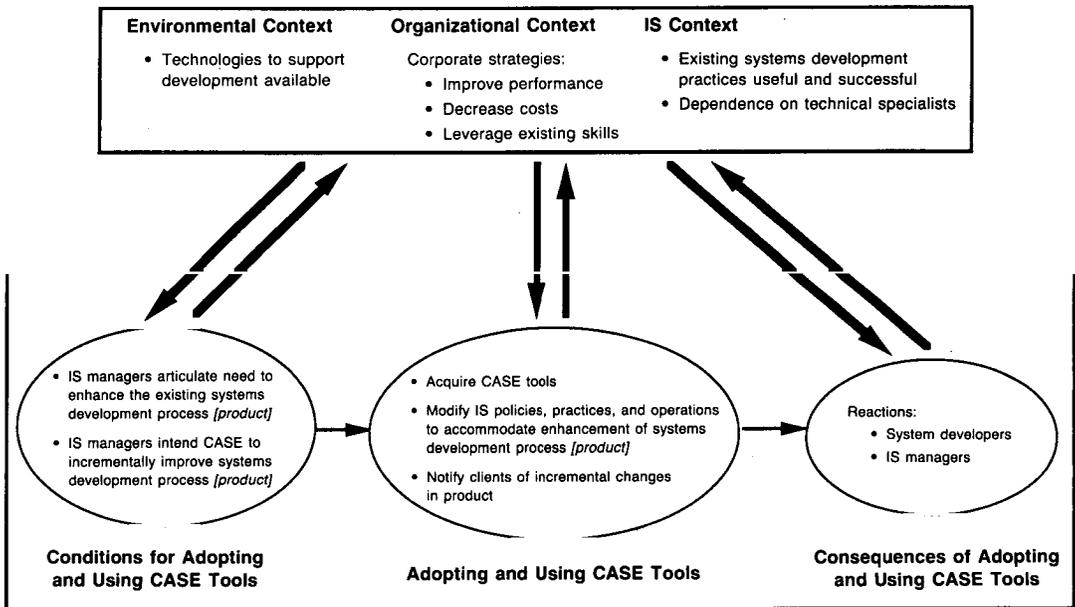
**Implications for systems development research**

The framework and findings discussed in this paper suggest that the contradictory results evident in the CASE literature likely reflect the fact that differences in intentions, processes, and contexts around the adoption and use of CASE tools

are largely overlooked by research that seeks invariant outcomes from CASE tools. In this, the framework proposed here is different from existing frameworks on technology adoption and implementation. These frameworks tend to share three characteristics: (1) they consist of causal models based on the deterministic assumptions of either a technological or organizational imperative (Markus and Robey, 1988) and therefore discount the importance of human intentions and action in shaping the adoption and use of technology; (2) they are variance models (Markus and Robey, 1988) and thus do not adequately capture the contextual and processual issues that are fundamental to examining organizational change (Dutton and Dukerich, 1991; Pettigrew, 1990); and (3) they focus primarily on activities of development and implementation and hence do not examine the use or consequences of a new technology.

While more empirical work is necessary to elaborate and verify the framework, it is believed that a useful starting point has been made.

**INSTITUTIONAL CONTEXT FOR ADOPTING AND USING CASE TOOLS**



**STRATEGIC CONDUCT IN ADOPTING AND USING CASE TOOLS**

Note: Items in italic font apply to Product Variations only. Items in normal font apply to both Product Variations and Process Variations.

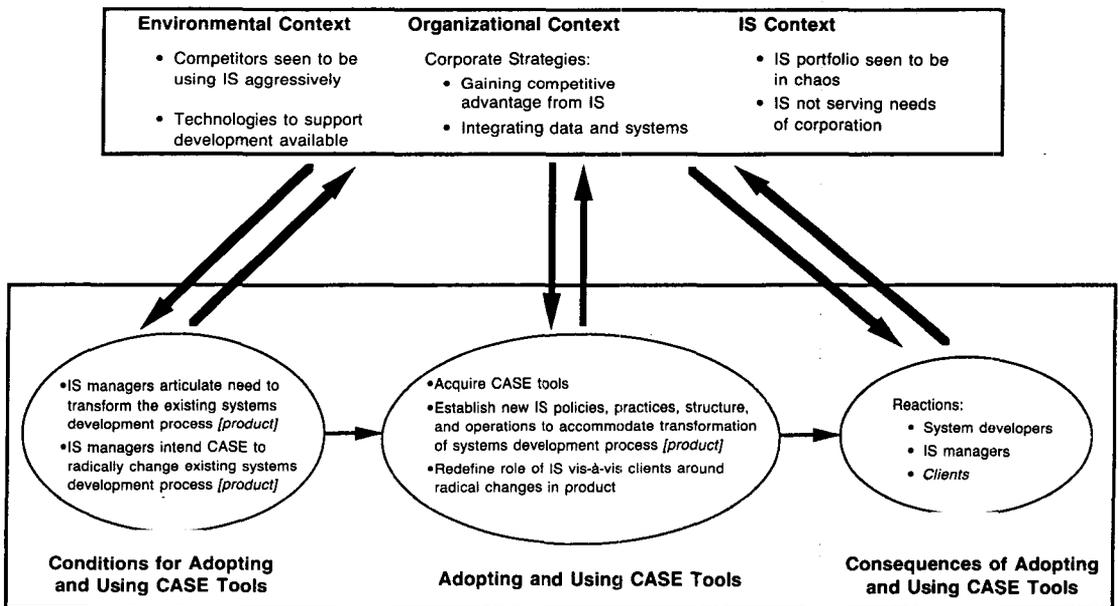
**Figure 3. Process of Organizational Change Around CASE Tools: An Illustration of Incremental Change**

Understanding the process of change around CASE tools (as depicted in Figure 1) and distinguishing between the nature of change and its locus of influence (as depicted in Figure 2) allow us to explain why two companies—on the surface introducing similar technology (integrated CASE tools)—should experience such different organizational changes. The framework suggests that to understand the adoption and use of CASE tools and their influence on systems development, it is necessary to examine the intentions and actions of key players, the social context into which the CASE tools are being implemented, and the change processes enacted as a result. Figure 1 details the kind of concepts that usefully capture these elements and their interaction over time. Further, the distinctions drawn between the locus of the technological change and its nature focus attention on where the influence of the CASE tools is being experienced and what challenges are being posed to the existing process and product of systems development.

Empirical validation and elaboration of these concepts in other settings are clearly needed. The theoretical framework was generated by only examining two sites, albeit in depth. More empirical grounding and comparisons will sharpen and enrich the concepts developed here and yield more complex understanding of the phenomenon. Three initial strategies for further research can be proposed.

First, it is necessary to investigate different contexts where CASE tools have been introduced. While the two companies studied here differed significantly on environment, strategy, size, structure, and culture, they still only represent two organizational types. More organizations need to be examined to see whether the proposed concepts and framework are relevant in other situations. In this way, the analytic generalization posited here—that other organizations' experiences with CASE tools will resemble the patterns detailed above—will be tested and elaborated.

**INSTITUTIONAL CONTEXT FOR ADOPTING AND USING CASE TOOLS**



**STRATEGIC CONDUCT IN ADOPTING AND USING CASE TOOLS**

**Note:** Items in italic font apply to Product Variations only. Items in normal font apply to both Product Variations and Process Variations.

**Figure 4. Process of Organizational Change Around CASE Tools: An Illustration of Radical Change**

Second, some of the dimensions in the Figure 2 grid may need to be elaborated or refined. While a dichotomous distinction in change intentions was adequate for this study, it is possible that future empirical work will require extending the dimensions. For example, some researchers have proposed finer-grained distinctions in types of innovation (Bartunek and Moch, 1987; Henderson and Clark, 1990; Meyerson and Martin, 1987), recognizing three or four rather than two types of change. Likewise, the distinctions between the process and product of systems development may not always be clear-cut, and a more finely calibrated classification of the systems development domain may be needed, for example, to distinguish between development and maintenance activities or between the development of mainframe applications and client-server applications.

Third, other combinations of positions on the Figure 2 grid should also be studied to find out the organizational consequences of hybrid changes. Both SCC and PCC adopted systems development changes that were consistent across process and product. It is conceivable, however, to think of organizations picking diagonal positions on the grid. Consider the following:

- An organization implements CASE tools to achieve process reorientations and product variations. This might occur, for example, when the IS unit continues to deliver stand-alone products to separate functional areas but transforms its process from one based on structured systems design to one using the object-oriented paradigm.
- An organization implements CASE tools to achieve process variations and product reorientations. This might occur, for example, when an existing process based on IE methodology is supplemented by the acquisition of IE-based CASE tools, and the IS unit decides to use the automated aids to implement a strategic IS plan and build a corporate information architecture with shared applications.

Empirical research into such experiences will help to establish the particular conditions, intentions, actions, and consequences associated with hybrid change strategies.

### *Implications for systems development practice*

The theoretical framework developed here meets the criteria of practical applicability proposed by Glaser and Strauss (1967). First, it fits the substantive area of study. The concepts and relations posited as central are intimately related to (because they are derived from) the arena of actual CASE tools adoption and use. Second, the theoretical framework is sufficiently general to be applicable to a range of situations around the adoption and use of CASE tools. It covers not only the process of change but also the location and type of change intended and experienced. Third, it is readily understandable by practitioners and should consequently provide some useful guidance in the actual change management plans of organizations introducing CASE tools. By providing practitioners with some insight into the context, structure, and process of CASE tools implementations, the framework serves as a basis from which the IS practitioner can assess and manage what is typically a poorly understood, complex, and dynamic situation.

The framework developed and presented here has important implications for IS practitioners. It suggests that before the implementation of a technology such as CASE tools, key players in the organization should articulate their intentions with respect to the tools and assess the context of systems development. Doing so will significantly influence the change process and resultant organizational consequences. Having examined and articulated these issues, key players can more effectively plan the implementation of CASE tools and facilitate the action required to enact the intended changes. The ensuing process will further shape the reactions, appropriations, and consequences of the CASE tools.

With respect to the impact of CASE tools on system developers, the findings suggest that it is not CASE tools alone that determine the reaction of the system developers. Rather, three types of attributes—individual, organizational, and technological—appear to significantly influence system developers' response to CASE tools (see Table 4). While these findings are from only two organizational sites and require further investigation, they do have considerable face validity because many anecdotal reports have hinted at one or other such influence.

**Table 4. Individual, Organizational, and Technological Influences on System Developers' Reactions to CASE Tools**

Category	Facilitating Characteristics	Constraining Characteristics
<b>Attributes of Individuals</b> Systems Experience  Time in Systems Development Career Orientation	Low investment in traditional systems development knowledge and practices  Less Business	Heavy investment in traditional systems development knowledge and practices  More Technical
<b>Attributes of Organization</b> Strategy Behind CASE Tools   Human Resource Policies   Career Paths	Improve systems and data quality Enhance skills and practices  Supportive of individual growth and development Job security Use of contract workers Multiple, alternative career paths	Decrease costs  Increase productivity Decrease reliance on system developers Not supportive of individual growth and development Limited job security  Single, competitive career path
<b>Attributes of Technology</b> Market Penetration	High	Low (e.g., built in-house)

In general, where system developers are not committed to a long-term career in IS, the implementation of CASE tools is unlikely to be seriously troubling to them. They may even welcome the tools' hiding of technical details and facilitation of business analysis. System developers who seek IS careers and value their technical skills may be distressed by the implementation of CASE tools. They may fear the obsolescence of their skills and object to the decreased involvement in the details of their work. This concern appears to be particularly acute when the tools are implemented as cost-cutting or productivity-improving measures. As we saw in the case of PCC, however, such concerns may be addressed by supportive human resource policies that provide adequate training and incentives to encourage skeptical or uncertain developers to learn and use the tools (e.g., by reassuring them that their jobs are not at risk or by not punishing initial poor performance) and that endorse alternative career paths (e.g., allow-

ing technically oriented developers to move into more technical areas, such as systems programming, database management, or CASE tools support).

The training and incentives given to system developers should be tailored to the type of change attempted. Incremental change, which builds on existing skills, work practices, and norms, requires programs and policies that reinforce the existing cognitive and social processes. Radical change, which departs from existing skills, work practices, and norms, requires programs and policies that foster the acquisition of new cognitive and social processes. While radical change offers the opportunity to implement a new vision and create fundamental change, it also invites greater risk and difficulty. Because it can be quite disruptive, radical change is often associated with ambiguity and chaos (Bartunek, 1984; Gash and Orlikowski, 1991; Hedberg, 1981; Tushman and Romanelli, 1985) and usually

is only attempted when managers perceive a rare opportunity or a crisis (as occurred in PCC).

Reorientations in systems development require IS managers to engender significant structural, procedural, and cultural change in their own unit and, in the case of product reorientations, throughout the organization. Radical change implies a paradigm shift, which requires a reframing and renegotiation of the IS mission, role, and relationships in the organization. With process reorientations, it is the system developers and IS managers—primarily involved in the IS work practices, structures, and skills—who are most directly affected. In such cases, senior IS managers need to champion and implement the radical changes in IS structure, knowledge, and work practices. Because there was no established, standardized systems development process in PCC prior to the adoption of CASE tools, the reorientation did not have to challenge long-standing norms or overcome deeply entrenched systems development practices.

When the IS product is changed, it is not only the IS unit but also the clients who will be affected. Product reorientations, thus, require senior and middle managers of the business units to champion and motivate the organizational changes required around fundamentally different IS products. As primary users of such products, business units may need to change their business processes, policies, and norms to take advantage of radically different information systems. To the extent that business managers do not recognize or accept the paradigm shift implicit in a product reorientation, as was evident in the case of PCC, the influence of CASE tools will be limited to the IS unit. Because PCC's practice of developing stand-alone IS products for separate business units was firmly institutionalized in PCC's decentralized culture, the shift to a corporate information architecture with shared data and applications proved to be a significant challenge. The business units' refusal to change as a result of the attempted IS product reorientation prevented any fundamental changes in the business. While these findings do not show that structural, procedural, and cultural changes by business units will lead to the successful adoption of IS product reorientations, they do suggest that where such changes are absent there will be significant problems of inertia, territorialism, and resistance.

This paper has argued that the implementation of CASE tools within systems development creates organizational change and that much can be gained by researching and managing it accordingly. While all research methodologies have strengths and weaknesses, the use of the grounded theory approach here was particularly appropriate, generating a set of insights, concepts, and interactions that address the critical organizational elements involved in adopting and using CASE tools, elements that to date have been largely overlooked in the CASE literature. The theoretical framework generated from the empirical findings suggests that the intentions and actions of key players, the change process they enact, as well as the social context into which tools are implemented, critically influence what changes are associated with CASE tools. This proposes a different approach to doing CASE tools research, one that takes into account the interaction over time of intentions, context, process, and action around the technology. Both researchers and practitioners alike should find the framework useful as they examine the implementation and use of this important but still poorly understood technology.

### Acknowledgements

The support of MIT's Center for Information Systems Research is gratefully acknowledged. Thanks are also due to Debra Hofman for many helpful discussions, to Jack Baroudi, Chris Kemerer, and Marcie Tyre for comments on an earlier draft of this paper, and to the associate editor and anonymous reviewers for their suggestions.

### References

- Agar, M.H. *The Professional Stranger: An Informal Introduction to Ethnography*, Academic Press, New York, NY, 1980.
- Ancona, D. "Outward Bound: Strategies for Team Survival in an Organization," *Academy of Management Journal* (33:2), June 1990, pp. 334-365.
- Banker, R.D. and Kauffman, R.J. "Reuse and Productivity in Integrated Computer-Aided Software Engineering: An Empirical Study," *MIS Quarterly* (15:3), September 1991, pp. 375-401.

- Bartunek, J. "Changing Interpretive Schemes and Organizational Restructuring: The Example of a Religious Order," *Administrative Science Quarterly* (29:3), September 1984, pp. 355-372.
- Bartunek, J. and Moch, M. "First Order, Second Order, and Third Order Change and Organization Development Interventions: A Cognitive Approach," *Journal of Applied Behavioral Science* (23:4), November 1987, pp. 483-500.
- Boland, R.J., Jr. "Control, Causality and Information System Requirements," *Accounting, Organizations and Society* (4:4), 1979, pp. 259-272.
- Boland, R.J., Jr. "Phenomenology: A Preferred Approach to Research on Information Systems," in *Research Methods in Information Systems*, E. Mumford, R. Hirschheim, G. Fitzgerald, and T. Wood-Harper (eds.), North-Holland, New York, NY, 1985, pp. 193-201.
- Card, D.N., McGarry, F.E., and Page, G.T. "Evaluating Software Engineering Technologies," *IEEE Transactions on Software Engineering* (SE-13:7), July 1987, pp. 845-851.
- Chua, W.F. "Radical Developments in Accounting Thought," *The Accounting Review* (LXI:4), October 1986, pp. 601-632.
- DeSanctis, G. and Poole, M.S. "Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory," *Organization Science*, forthcoming.
- Dewar, R.D. and Dutton, J.E. "The Adoption of Radical and Incremental Changes: An Empirical Analysis," *Management Science* (32:11), November 1986, pp. 1422-1433.
- Dutton, J.E. and Dukerich, J.M. "Keeping an Eye on the Mirror: Image and Identity in Organizational Adaptation," *Academy of Management Journal* (34:3), September 1991, pp. 517-554.
- Eisenhardt, K.M. "Building Theories from Case Study Research," *Academy of Management Review* (14:4), October 1989, pp. 532-550.
- Elsbach, K.D. and Sutton, R.I. "Acquiring Organizational Legitimacy through Illegitimate Actions: A Marriage of Institutional and Impression Management Theories," *Academy of Management Journal* (35:4), December 1992, pp. 699-738.
- Ettlie, J., Bridges, W., and O'Keefe, R. "Organization Strategy and Structural Differences for Radical versus Incremental Innovation," *Management Science* (30:6), June 1984, pp. 682-695.
- Fichman, R. and Kemerer, C. "Adoption of Software Engineering Process Innovations: The Case of Object Orientation," *Sloan Management Review* (34:2), Winter 1993, pp. 7-22.
- Gash, D.C. and Orlikowski, W.J. "Changing Frames: Towards an Understanding of Information Technology and Organizational Change," in *Academy of Management Best Papers Proceedings, 51st Annual Meeting*, J.L. Wall and L.R. Jauch (eds.), Academy of Management, Ada, OH, 1991, pp. 189-193.
- George, J.F. and King, J.L. "Examining the Computing and Centralization Debate," *Communications of the ACM* (34:7), July 1991, pp. 63-72.
- Ginzberg, M.J. "Early Diagnosis of MIS Implementation Failure: Promising Results and Unanswered Questions," *Management Science* (27:4), April 1981, pp. 459-478.
- Glaser, B.G. and Strauss, A.L. *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Aldine Publishing Company, New York, NY, 1967.
- Hedberg, B. "How Organizations Learn and Unlearn," in *Handbook of Organizational Design*, P.C. Nystrom and W.H. Starbuck (eds.), Oxford University Press, Oxford, United Kingdom, 1981, pp. 3-27.
- Henderson, R.M. and Clark, K.M. "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms," *Administrative Science Quarterly* (35:1), March 1990, pp. 9-30.
- Isabella, L.A. "Evolving Interpretations as a Change Unfolds: How Managers Construe Key Organizational Events," *Academy of Management Journal* (33:1), March 1990, pp. 7-41.
- Kahn, W.A. "Psychological Conditions of Personal Engagement and Disengagement at Work," *Academy of Management Journal* (33:4), December 1990, pp. 692-724.
- Leonard-Barton, D.A. "A Dual Methodology for Case Studies: Synergistic Use of a Longitudinal Single Site with Replicated Multiple Sites," *Organization Science* (1:3), August 1990, pp. 248-266.
- Lucas, H.C., Jr. "Empirical Evidence for a Descriptive Model of Implementation," *MIS Quarterly* (2:2), June 1978, pp. 27-42.
- Markus, M.L. "Power, Politics, and MIS Im-

- plementation." *Communications of the ACM* (26:6), June 1983, pp. 430-444.
- Markus, M.L. and Robey, D. "Information Technology and Organizational Change: Causal Structure in Theory and Research," *Management Science* (34:5), May 1988, pp. 583-598.
- Martin, J. *Information Engineering: Book I Introduction*, Prentice Hall, Englewood Cliffs, NJ, 1990a.
- Martin, J. *Information Engineering: Book II Planning and Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1990b.
- Martin, J. *Information Engineering: Book III Design and Construction*, Prentice Hall, Englewood Cliffs, NJ, 1990c.
- Martin, P.Y. and Turner, B.A. "Grounded Theory and Organizational Research," *The Journal of Applied Behavioral Science* (22:2), 1986, pp. 141-157.
- Meyerson, D. and Martin, J. "Cultural Change: An Integration of Three Different Views," *Journal of Management Studies* (24:6), November 1987, pp. 623-647.
- Miles, M.B. and Huberman, A.M. *Qualitative Data Analysis: A Sourcebook of New Methods*, Sage Publications, Newbury Park, CA, 1984.
- Mohr, L.B. *Explaining Organizational Behavior*, Jossey Bass, San Francisco, CA, 1982.
- Necco, C., Tsai, N.W., and Holgeson, K.W. "Current Usage of CASE Software," *Journal of Systems Management* (40:5), May 1989, pp. 6-11.
- Norman, R. and Nunamaker, J.F. "An Empirical Study of Information Systems Professionals' Productivity Perceptions of CASE Technology," in *Proceedings of the Ninth International Conference on Information Systems*, J.I. DeGross and M.H. Olson (eds.), copies available from Association for Computing Machinery, New York, NY, 1988, pp. 111-118.
- Norman, R., Corbitt, G., Butler, M., and McElroy, D. "CASE Technology Transfer: A Case Study of Unsuccessful Change," *Journal of Systems Management* (40:5), May 1989, pp. 33-37.
- Normann, R. "Organizational Innovativeness: Product Variation and Reorientations," *Administrative Science Quarterly* (16:2), June 1971, pp. 203-215.
- Orlikowski, W.J. *Information Technology and Post-Industrial Organizations: An Examination of the Computer-Mediation of Production Work*, unpublished doctoral dissertation, Stern School of Business, New York University, New York, NY, 1988a.
- Orlikowski, W.J. "CASE Tools and the IS Workplace: Some Findings From Empirical Research," in *Proceedings of the ACM SIGCPR Conference*, E.M. Awad (ed.), ACM Press, Baltimore, MD, April 1988b, pp. 88-97.
- Orlikowski, W.J. "Division Among the Ranks: The Social Implications of CASE Tools for Systems Developers," in *Proceedings of the Tenth International Conference on Information Systems*, J.I. DeGross, J.C. Henderson, and B.R. Konsynski (eds.), copies available from Association for Computing Machinery, New York, NY, 1989, pp. 199-210.
- Orlikowski, W.J. "Integrated Information Environment or Matrix of Control? The Contradictory Implications of Information Technology," *Accounting, Management, and Information Technologies* (1:1), 1991, pp. 9-42.
- Orlikowski, W.J. "The Duality of Technology: Rethinking the Concept of Technology in Organizations," *Organization Science* (3:3), August 1992, pp. 398-427.
- Orlikowski, W.J. and Baroudi, J.J. "Studying Information Technology in Organizations: Research Approaches and Assumptions," *Information Systems Research* (2:1), March 1991, pp. 1-28.
- Orlikowski, W.J. and Robey, D. "Information Technology and the Structuring of Organizations," *Information Systems Research* (2:2), June 1991, pp. 143-169.
- Pennings, J. "Information Technology in Production Organizations," *International Studies of Management and Organization* (17:4), 1988, pp. 68-89.
- Pettigrew, A.M. "Issues of Time and Site Selection in Longitudinal Research on Change," in *The Information Systems Research Challenge: Qualitative Research Methods*, J.I. Cash, Jr. and P.R. Lawrence (eds.), Harvard Business School Press, Boston, MA, 1989, pp. 13-19.
- Pettigrew, A.M. "Longitudinal Field Research on Change: Theory and Practice," *Organization Science* (1:3), August 1990, pp. 267-292.
- Rogers, E. *The Diffusion of Innovation*, Free Press, New York, NY, 1983.
- Strauss, A. and Corbin, J. *Basics of Qualitative Research: Grounded Theory, Procedures, and Techniques*, Sage Publications, Newbury

- Park, CA, 1990.
- Sutton, R.I. "The Process of Organizational Death: Disbanding and Reconnecting," *Administrative Science Quarterly* (32:4), December 1987, pp. 542-569.
- Swanson, K., McComb, D., Smith, J., and McCubbrey, D. "The Application Software Factory: Applying Total Quality Techniques to Systems Development," *MIS Quarterly* (15:4), December 1991, pp. 567-579.
- Turner, B.A. "The Use of Grounded Theory for the Qualitative Analysis of Organizational Behaviour," *Journal of Management Studies* (20:3), July 1983, pp. 333-348.
- Tushman, M.L. and Romanelli, E. "Organizational Evolution: A Metamorphosis Model of Convergence and Reorientation," in *Research in Organizational Behavior*, L.L. Cummings and B.M. Staw (eds.), JAI Press, Greenwich, CT, 1985, pp. 171-222.
- Vessey, I., Jarvenpaa, S.L., and Tractinsky, N. "Evaluation of Vendor Products: CASE Tools as Methodology Companions," *Communications of the ACM* (35:4), April 1992, pp. 90-105.
- Yellen, R. "Systems Analysts Performance Using CASE versus Manual Methods," in *Proceedings of the Twenty-Third Annual Hawaii International Conference on System Sciences*, IEEE Computer Society Press, Los Alamitos, CA, 1990.
- Yin, R.K. *Case Study Research: Design and Methods*, Sage Publications, Beverly Hills, CA, 1989a.
- Yin, R.K. "Research Design Issues in Using the Case Study Method to Study Management Information Systems," in *The Information Systems Research Challenge: Qualitative Research Methods*, J.I. Cash, Jr. and P.R. Lawrence (eds.), Harvard Business School Press, Boston, MA, 1989b, pp. 1-6.
- Yourdon, E. and Constantine, L.L. *Structured Design*, Yourdon Press, New York, NY, 1978.

### About the Author

**Wanda J. Orlikowski** is assistant professor of information technologies in the Sloan School of Management at the Massachusetts Institute of Technology. She received her Ph.D. in information systems from New York University. Her primary research interest focuses on the interaction between information technologies and organizational dimensions, such as structure, culture, control mechanisms, and work processes. She is currently investigating the process of organizational change around the adoption and use of coordination technologies. Her articles have appeared in *Academy of Management Review*, *Information Systems Research*, *MIS Quarterly*, and *Organization Science*. Dr. Orlikowski serves on the editorial boards of *Accounting, Management and Information Technologies*, *Information Technology & People*, *MIS Quarterly*, and *Organization Science*.