Paul Nelson,
William Richmond,
and Abraham Seidmann

Two Dimensions of Software Acquisition

SOFTWARE acquisition is a critical concern for firms worldwide, and its impact on their business processes is expected to grow. U.S. firms spend more than $250 billion annually acquiring software [18]. Software investments can result in substantial productivity gains and strategic advantages [2], but not all such investments are successful by any means [3]. To realize these benefits, firms must identify and understand the factors that most affect their software acquisition decisions.

In this article, we develop a general cost-benefit decision framework for the software acquisition problem. We use this framework and software acquisition data to investigate six hypotheses characterizing software acquisition. Our framework and empirical results offer managers a basis for structuring and benchmarking their software acquisition decisions and offer vendors insights into the characteristics of the software projects that would be most beneficial to target.

The software acquisition problem is depicted as two-dimensional (2D)—what acquisition approach to use (custom develop the software or base it on a package) and who should complete the task (internal resources or external providers). While aspects of the sourcing decision have generated frequent study, the custom/package decision has not previously received attention. In addition, our decision framework captures the major tradeoffs a firm faces when making these two decisions.

Previous theoretical studies have focused on singular aspects of the insource/outsource decision. Our framework unites these aspects. It builds upon the contracting and monitoring cost issues in the transaction-cost and incomplete-contracting literatures and includes the tradeoff between monitoring and production costs developed in the agency theory literature [13, 17].

The model also encompasses the various managerial rationales provided in the extensive empirical literature regarding firms' sourcing decisions for information technology (IT) services [8, 12] and the limited literature on software acquisition [16]. Our analysis extends this literature by using firm actions, not post-decision rationalizations, to determine the key factors behind acquisition decisions. Following findings in the marketing literature [5], relative factor weights inferred from actual decision data should provide greater explanatory power than previous studies have offered.

Software Acquisition Cost-Benefit Framework

The 2D, four-option sourcing decision is depicted in Table 1. Using packaged software involves identifying the company's information processing requirements, evaluating alternative packages, selecting one, possi-

New field data supports the contention that two dimensions of the software acquisition problem—custom/package and insource/outsource—are interrelated at several levels.

Table 1. The software acquisition problem

Acquisition Approach	Acquisition Team: Insource	Acquisition Team: Outsource
Custom	Internal resources only for needs analysis, coding, etc.	Vendor performs needs analysis, coding, etc.
Package	Internal resources only for package selection, installation, etc.	Vendor performs package selection, installation, etc.

bly modifying it, and installing and testing the selected system. If an external provider carries out these functions, the acquisition is an *outsourced package* (denoted {outsource, package}). If in-house staff are used, the acquisition is an *insourced package*, {insource, package}.

Customized software is developed when it is not cost-beneficial to meet the user's requirements through modifications to packaged software. Custom software acquisition generally involves a series of possibly repeated steps: needs analysis, system design, coding, testing, training, and installation. Outsourced custom projects {outsource, custom} use vendors to complete these activities; insourced custom development {insource, custom} uses only internal resources.

The problem facing the firm is to identify the acquisition option that will maximize the net present value of the software acquisition, subject to any organizational considerations (e.g., corporate policy and resource availability). To achieve this goal, for each available acquisition option, the firm must determine the expected benefits of the resulting software system (*system value*) and a variety of acquisition-related costs. These costs include production costs [*needs analysis coding, and installation*], *monitoring costs,* and *contracting costs.*

The financial magnitudes of the cost-benefit drivers differ depending upon whether an internal or external software acquisition team is used and whether the software is based on a package or is custom developed. Table 2 presents the cost-benefit drivers and identifies which drivers typically favor a particular acquisition team or approach.[1]

A particular software system is designed to meet requirements that support a set of business functions (e.g., customer reservations and manufacturing resource planning). When comparing internal and external acquisition, system value does not differ, because both teams are expected to deliver the same system. A custom system should have greater value, since it can be specified to better fit application requirements.

The requirements necessary to automate the desired business functions are determined by needs analysis. These costs are lower internally, because an internal information systems group better understands the firm's business operations. Needs analysis costs are the same on the custom/package dimension, because both require the same analysis.

Coding and installation (*code and install*) includes: the detailed software design that translates the users' requirements into a programmable solution; the coding, testing, and installation of the programs; and the conversion of business operations over to the new system. For software packages, the design and coding and some of the testing costs are included in the purchase price or licensing fee. Coding and installation costs are higher when software is developed internally, because external markets have more competitive sources of labor and technical expertise, and they are in a better position to take advantage of scale and scope economies [8]. A similar argument holds for packaged software.

The quality of completed software is difficult to assess without extended use, and only imprecise evidence of development progress is available.[2] This heightens concerns over opportunistic behavior by employees or vendors. Monitoring and contracting act as controls on these concerns. Monitoring costs include project management costs and any costs for quality assurance, progress reviews, and training. These costs are lower with external development. Largely due to scale and scope economies and process change management, vendors manage software production activities more efficiently [7]. Monitoring costs are lower for packaged software because of its *physical good* aspects.

There are also several contracting costs: searching for and evaluating potential custom or packaged software vendors, benchmarking and screening their capabilities, specifying the legal terms of a contract, negotiating the contract, and resolving disputes. These costs are lower for internal acquisition, because the company does not have to search for a vendor, negotiate a detailed contract, or use external entities to

Table 2. Impact of cost-benefit drivers on the software acquisition decision

	Acquisition team		Acquisition approach	
Cost-benefit Driver	Insource	Outsource	Custom	Package
System value			↑	
Needs analysis	↑			
Code and install		↑		↑
Monitoring		↑		↑
Contracting	↑			↑

Upward-pointed arrows (↑) denote which cost-benefit drivers typically favor different acquisition teams or approaches

[1]These generalizations are based on findings in the academic literature and managerial statements.

[2]That is, software is not a search good. Rather, repeated experiences with the product are required to fully ascertain its quality (i.e., it is an experience or credence good) [14].

enforce it. Instead, monitoring is typically used for internal personnel. Contracting costs are lower for packaged software because of its physical good aspects.

Hypotheses

Generally, for a particular software acquisition decision, the financial values of the cost-benefit drivers for each acquisition option are unavailable to managers prior to an acquisition decision. But information which *is* available to them includes: the characteristics of the desired software and application to be automated, the firm's abilities relative to these characteristics, and the organizational attitudes and constraints concerning software acquisition. Therefore, the decision process must center on three observable factors: technological features, application properties, and organizational considerations. The first two directly affect the cost-benefit drivers and hence predictably steer managerial decisions. Organizational considerations also influence the decision, although they may or may not affect the cost-benefit drivers directly.

Table 3 outlines the intuitive relationships among these three factors, the five cost-benefit drivers, and the two dimensions of the software acquisition decision. From these relationships, four hypotheses addressing the expected marginal impacts of technological, application, and organizational factors on a particular software acquisition decision are generated. Hypotheses concerning a temporal effect and the interdependence of the two dimensions are also proposed.

Hypothesis 1 (technological features): *Systems using specialized technology or advanced development environments are, all else equal, more likely to be outsourced and/or associated with packaged software.*

A primary reason given for outsourcing is the acquisition of cost-effective technological expertise [16]. As the desired technology becomes more specialized or advanced, coding and installation costs more strongly favor outsourcing. Complex technology also increases monitoring and contracting costs. While the relative increase in the importance of monitoring costs is likely to be smaller than that for contracting costs, outsourcing's increased coding and installation cost advantage is likely to dominate, thus favoring outsourcing. Packaged software is favored due to the increased importance of coding and installation, monitoring, and contracting costs.

Table 3. Relationships among project characteristics, cost-benefit drivers, and software acquisition

Hypotheses	Project characteristics	Relative importance of cost-benefit drivers					Software acquisition: Acquisition team		Software acquisition: Acquisition approach	
		System Value	Needs Analysis	Code and Install	Monitoring	Contracting	Insource	Outsource	Custom	Package
H1	Specialized/ Advanced Technology			↑	↑	↑		↑		↑
H2	Strategic Application	↑	↑		↑	↑	↑		↑	
H3	Common Application			↑				↑		↑
H4	Organizational Considerations						↑		↑	
H5	Installation Date After 1990							↑		

On the left side of the table, upward-pointed arrows (↑) denote those cost-benefit drivers whose relative importances are increased by a particular project characteristic. Based on the discussion pertaining to Table 1 on how the cost-benefit drivers affect the axquisition team and approach decisions, the arrows on the right side of the table denote which acquisition team and approach are more likely as a result of each particular project characteristic. Thus, the right side of the table synopsizes Hypotheses 1 through 5 (H_1–H_5).

Hypothesis 2 (application properties): *Strategic applications are more likely to be insourced and/or custom-developed, all else equal.*

Hypothesis 3 (application properties): *Common applications are more likely to be outsourced and/or acquired as packaged software, all else equal.*

Two application properties—the application's uniqueness and its strategic role—are commonly cited as favoring insourcing [8]. Strategic applications are intended to provide a competitive advantage. Consequently, needs analysis, monitoring, and contracting costs become relatively more important. For such an application, the higher contracting costs for outsourcing are expected to exceed insourcing's higher monitoring costs. This, coupled with the increased importance of needs analysis, favors the use of internal acquisition. Since the effect on system value is expected to be stronger than the effect on potential opportunistic behavior, strategic applications are more likely to be custom developed. For applications that are common to many organizations

and/or that use standard data structures, vendors can leverage their resources across multiple clients. This increases the relative advantage of outsourcing and packaged software with respect to coding and installation costs.

Hypothesis 4 (organizational considerations): *Organizational considerations favor insourced and/or custom software acquisition, all else equal.*

A company's culture and the system's sponsor have been identified as significant factors in software acquisition decisions [1]. Examples of possible organizational biases are the "not-invented-here" syndrome and "kingdom building" [6]. Both result in a tendency toward insourced, custom acquisition. Alternatively, a firm or system sponsor may have a predisposition toward outsourcing or packaged software. Organizational considerations of this type can be seen as covariates to the cost-benefit model.

Hypothesis 5 (date of installation): *Outsourced software acquisition has become more likely over time, all else equal.*

Recent publications perceive an increase in the level of outsourcing over time [10].

Hypothesis 6 (interaction between the custom/package and insource/outsource decisions): *An interaction exists between the insource/outsource and custom/package decisions.*

The software acquisition problem is fundamentally different from the classic manufacturer's make/buy decision that refers to the repetitive acquisition of a packaged good (input). That is, the problem involves how to make an input, as well as who should make it. In addition, these two decisions interact. The confounding effects of each decision on the other lead the firm to combine these decisions, rather than make them separately. For example, the use of packaged software lowers the relative importance (monetary value) of all the drivers except needs analysis costs, with coding and installation costs the most affected. Consequently, the use of a package favors insourcing.

Data

Survey data was collected for every documented software acquisition project undertaken by each of five firms. The data covers 186 system projects ranging from small personal computer applications to large, distributed systems and from common accounting systems to strategic operational systems; the systems were installed between 1967 and 1993. The annual revenues of the participating companies ranged from $20 million to $900 million. These companies included a regulated utility, a long-distance telephone company, a payroll processing company, a rocket engine manufacturer, and a building products manufacturer.

For each project, the firm's actions—not post-decision rationalizations—were ascertained concerning both dimensions of the software acquisition decision: 64% were {insource, custom} projects, 17% were {outsource, custom}, 11% were {insource, package}, and the remaining 8% were {outsource, package}.

Five technological features for each system were noted:

- database management system(s) (DBMS),
- programming language(s),
- hardware platform(s),
- system architecture, and
- processing mode.

Four variables were ascertained to measure each system's application properties:

Table 4. Four-option multinominal logit results

Project Characteristics	Variables	{Outsource, Custom} Parameters	{Insource, Package} Parameters	{Outsource, Package} Parameters
Technological Features	**H5**			
DBMS	File-based	---	---	–2.17***
Programming language	Third-generation	–0.70	---	---
	Fourth-generation	---	---	2.18***
Hardware platform	Minicomputer	2.01***	4.34***	3.63***
	Multiple	1.61***	1.81*	2.32**
System architecture	Distributed	---	---	---
Processing mode	Batch	2.22***	---	---
Application Properties	**H2,H3**			
Strategic mission	Strategic	---	0.98*	---
Application uniqueness	Common	0.89*	2.44***	2.77***
Application type	Transaction Processing	1.43**	---	---
	Combination	1.42*	---	1.61**
Functional area	Finance	---	3.14***	---
Organizational Considerations	**H4**			
Firm	Firm 2	22.80***	4.28***	5.63***
	Firm 3	---	---	3.88**
	Firm 4	22.80***	4.28***	5.63***
	Firm 5	22.80***	2.61**	5.63***
Sponsor level	Executive	---	---	---
Time of Installation	**H5**			
	Before 1981	–2.67***	---	---
	After 1990	---	---	---
Intercept Term	**H4**			
	Intercept	–25.73***	–9.91***	–10.47***
Goodness of Fit				
	Log Likelihood Value	–123.0***		
Significance Levels (One-Sided)	*** 1% level ** 5% level * 10% level			

Results pertain to a four-option version of the model expressed by equations (3) and (4). The parameter estimates provided relate to the likelihood of the three noted acquisition options relative to a base case, which is {insource, custom}.

- application type,
- functional area,
- application uniqueness, and
- strategic mission.

The first three measures pertain to the application's uniqueness; the fourth pertains to the application's strategic role. Two variables measure organizational considerations: *sponsor level* (the system sponsor's organizational level) and *firm*. The firm-specific dummy variables for each of the five firms are used to encompass idiosyncratic organizational considerations. The installation date of the system was recorded to measure a possible time trend.

Analysis and Results

We constructed three models to identify the existence and type of interaction between the custom/package and insource/outsource decisions. Underlying each model is the idea that the utility of an acquisition option depends on its perceived costs and benefits, which in turn depend on the system's technological, application, organizational, and temporal factors. This is similar to the traditional multiattribute utility framework used in marketing and economics to model brand choice with respect to a set of products or services [4]. For any project m, the utility of acquisition option j is

$$U_j = \alpha_j + \sum_i \beta_{ij} X_{im}. \quad (1)$$

The α_j intercept term is an option-specific constant depicting a firm's predisposition to use option j (i.e., organizational biases or corporate policy). The explanatory variables, X_{im}, measure the technological, application, organizational, and temporal factors. Given our data, X_{im} is a dummy binary variable equal to 1 if variable i is present in project m and equal to 0 otherwise. The β_{ij} terms reflect the impact of a particular explanatory variable i on the utility of option j.

The firm wishes to choose the acquisition option with the largest utility. However, due to unmodeled factors and random events, the estimate of each acquisition option's utility equals its true utility plus an error term, $U_j^{est} = U_j + \epsilon_j$. If these error terms are assumed to be distributed Gumbel, the multinomial logit model results. This model offers an intuitively appealing representation for the probability of choosing option j [11]:

$$P[j] = \frac{e^{u_j^{est}}}{\sum_{k=1}^{K} e^{u_k^{est}}} \quad (2)$$

For expository ease and identification reasons, we define option 1 as a base case with utility normalized to zero. For a particular project, the utilities of the other options then depict the differences in utility between that option and the base case. After some mathematical manipulation, the probability that option j is chosen for project m is

$$P[1] = \frac{1}{1 + \sum_{k=2}^{K} e^{\gamma_k + \sum_i \delta_{ik} X_{im}}} \quad (3)$$

$$P[j] = \frac{e^{\gamma_j + \sum_i \delta_{ij} X_{im}}}{1 + \sum_{k=2}^{K} e^{\gamma_k + \sum_i \delta_{ik} X_{im}}}, \; j = 2,3,\ldots,K, \quad (4)$$

where $\gamma_j = \alpha_j - \alpha_1$ and $\delta_{ij} = \beta_{ij} - \beta_{i1}$.

The first possible model of software acquisition is a simple four-option version of the multinomial logit model proposed in equations (3) and (4). The base case is defined as {insource, custom}. Option 2 is {outsource, custom}; option 3 is {insource, package}; option 4 is {outsource, package}.

The other two models are variants of this logit framework. A simpler model assumes that decisions on the two dimensions of the problem are independent. A more general model allows for a hierarchical (nested) interaction in which the sourcing decision is conditional on the custom/package decision.

Statistical evaluations showed that the four-option multinomial logit model best depicts firm behavior: An interaction does exist between the firm's custom/package and insource/outsource decisions (Hypothesis 6).[3] The estimation results for this model provided in Table 4 show significant support for Hypotheses 1 through 5. Note that some explanatory variables for particular options have been dropped in order to simplify the analysis, reduce collinearity problems, and focus attention on the key decision factors. If a variable was dropped, it was not statistically significant (one-sided) at the 10% level. A positive (negative) parameter value means the probability of the denoted acquisition option increases (decreases) relative to the base case {insource, custom}, holding all else constant. For example, the negative intercept terms imply that the base case {insource, custom} is more likely than any other option. The discussion of each parameter estimate that follows implicitly holds all else constant.

Technological Features

We find general support for the idea that systems using specialized technology or advanced development environments are more likely to be outsourced and/or associated with packaged software (Hypothesis 1). Parameter values for four of the five technological factors provide support for this hypothesis.

[3]Further details concerning the models, statistical analysis, and other aspects of this article may be found in [15].

Both the system's technological features and the application's properties play a significant role in the software acquisition decision.

DBMS: The {outsource, package} option is less likely when file-based systems are used. This follows our hypothesis, since these systems are simpler and do not require advanced development environments.

Programming language: Systems using fourth-generation languages are more likely to be {outsource, package}. Weak evidence exists that systems written in third-generation languages are less likely to be {outsource, custom}. These findings support Hypothesis 1. Third-generation languages are a standard technology, whereas fourth-generation languages are associated with advanced development environments.

Hardware platform: Applications running on multiple platforms or minicomputers are more likely to be associated with outsourcing and packaged software. Both findings support Hypothesis 1. Multiple platform systems require specialized abilities in systems integration. Minicomputer environments frequently encounter more binding technological expertise and manpower constraints.

System architecture: This has no statistically significant effect on the software acquisition decision.

Processing mode: Companies are more likely to outsource custom development of batch systems despite the low technological complexity. The custom aspect of this finding fits with Hypothesis 1, but the outsourcing component is unexpected.

Application Properties

There is little support for the hypothesis that strategic applications are more likely to be insourced and/or custom-developed (Hypothesis 2). There is weak statistical evidence that strategic applications are more likely to be {insource, package}. With respect to insourcing, this is consistent with Hypothesis 2, but the use of packaged software is counter to our expectations. We propose two explanations for these results:

- Some applications classified as strategic are mission-critical to firm operations but do not give the firm a strategic advantage.
- The information system itself does not generate a competitive advantage. Rather, the way the system is used generates the value. Examples are the CATIA and SAP software packages that support the core competencies of many competing firms.

These explanations are consistent with those made by Lacity, Willcocks, and Feeny [9] for why IT services labeled strategic are not necessarily insourced.

Our three application uniqueness variables strongly confirm that common applications are more likely to be based on a package and/or to be outsourced (Hypothesis 3).

Application type: Transaction processing systems are more likely to be {outsource, custom}. They support critical, high-volume functions common to many businesses, but each system requires substantial firm-specific coding in order to meet the firm's idiosyncratic data and processing requirements.

Table 5. Relationship between technological and application factors and the software acquisition decision

Application Factors	Technological Factors: Standard/Simple	Technological Factors: Specialized/Advanced
Common	PACKAGE insource or outsource?	OUTSOURCE custom or package?
Unique	INSOURCE custom or package?	CUSTOM insource or outsource?

Functional area: Applications in finance/accounting are strongly associated with {insource, package}. These applications (e.g., general ledger) are typically common.

Application uniqueness: Common applications are more likely to be outsourced and/or packaged.

The results for Hypotheses 1 and 3 are depicted in Table 5: Neither application nor technological factors dominate software acquisition decisions. A system addressing a common application and based on common technology is more likely to be acquired as a package, but there is no clear evidence concerning the sourcing dimension. Similarly, a system addressing a common application and based on specialized technology and advanced development environments is more likely be outsourced, but there is no clear evidence concerning the custom/package dimension. Analogous findings pertain to the other two quadrants.

Organizational Considerations and Temporal Effects

We find that organizationwide biases exist in all of the firms analyzed. There is also support for the idea that companies have idiosyncratic tendencies concerning their software acquisition decisions (Hypothesis 4). The intercept terms and firm-specific constants show that, possibly because of kingdom building or the not-invented-here syndrome, all firms have a strong predisposition toward {insource, custom}. Firms 2, 4, and 5 are less predisposed toward custom development, while firm 3 is less averse to packaged software if its installation is done by an outside vendor. On the other hand, the organizational level of the project's sponsor has no effect on either software acquisition decision dimension.

Consistent with Hypothesis 5, outsourced custom systems were less likely before the 1980s. No change is found concerning packaged software. This implies that any increase in the use of packaged software over time is captured by changes in technological, application, and organizational factors.

Research and Management Implications

This article is the first to develop a general economic framework addressing the software acquisition problem in a broader context that encompasses the custom/package and insource/outsource decisions. Using this framework, the key decision factors behind actual software acquisitions are identified using logit analysis. This extends previous studies that analyzed post-decision rationales.

Our key findings are as follows: Companies do not make the custom/package and insource/outsource decisions independently. Both the system's technological features and the application's properties play a significant role in the software acquisition decision. Neither always dominates the other. For example, all else equal, systems using specialized and advanced technologies to address common applications tend to be outsourced, and systems using simpler, more basic technologies to address common applications tend to be acquired as packages. Contrary to popular belief, strategic applications are not more likely, all else equal, to be insourced and/or custom developed. Companies have a strong predisposition to acquire software through internal, custom development. This predisposition with respect to insourcing has decreased over time; however, no temporal change in the adoption of packaged software is observed.

The framework developed and empirical findings facilitate a broader and more detailed understanding of the software acquisition problem. They give managers a basis for structuring and benchmarking software acquisition decisions, they provide vendors with insights into the types of software projects they should target, and they offer academics a stepping stone to more focused research on particular aspects of the problem, such as bundle pricing, incentives, warranties, support services, managerial oversight, and benchmarking. ◻

References

1. Arnet, K.P., and Jones, M.C. Firms that choose outsourcing: A profile. *Info. Manage. 26* (1994), 179–188.
2. Brynjolfsson, E. The productivity paradox of information technology. *Commun. ACM 36,* 12 (1993), 67–77.
3. Gibbs, W. Software's chronic crisis. *Sci. Am.* (Sept. 1994), 86–95.
4. Green, P, and Srinivasan, V. Conjoint analysis in marketing: New developments with implications for research and practice. *J. Mark 54* (1990), 3–19.
5. Hauser, J.R., and Urban, G.L. A normative methodology for modeling consumer response to innovation. *Oper. Res. 25* (1977), 579–619.
6. Jensen, M., and Meckling, W.H. Theory of the firm: managerial behavior, agency costs and ownership structure. *J. Financ. Econ. 3* (1976), 305–360.
7. Joch, A. How software doesn't work. *Byte* (Dec. 1995), 49–60.
8. Jones, C. Evaluating software outsourcing options. *Info. Sys. Manage. 11* (1994), 28–33.
9. Lacity, M., Willcocks, L., and Feeny, D. IT outsourcing: Maximizing flexibility and control. *Harvard Bus. Rev. 73* (1995), 84–93.
10. Loh, L. and Venkatraman, N. Determinants of information technology outsourcing: A cross-sectional analysis. *J. Manage. Info. Sys. 8* (1992), 7–24.
11. McFadden, D. Conditional logit analysis of qualitative choice behavior. In *Frontiers in Econometrics,* P. Zarembka, Ed. Academic Press, New York, 1974.
12. McFarlan, F.W., and Nolan, R.L. How to manage IT outsourcing alliance. *Sloan Manage. Rev. 35* (1995), 9–23.
13. Milgrom, P., and Roberts, J. *Economics, Organization and Management.* Prentice-Hall, Englewood Cliffs, N.J., 1992.
14. Nelson, P. Information and consumer behavior. *J. Polit. Econ. 78* (1970), 311–329.
15. Nelson, P., Richmond, W., and Seidmann, A. Software acquisition: The custom/package and insource/outsource dimensions. CIS Working Paper, University of Rochester, Rochester, New York, 1995.
16. Patane, J.R., and Jurison, J. Is global outsourcing diminishing the prospects for American programmers? *J. Sys. Manage. 45* (1994), 6–10.
17. Richmond, W.B., Seidmann, A., and Whinston, A. Incomplete contracting issues in information system development outsourcing. *Decis. Support Syst. 8* (1992), 459–477.
18. Standish Group. Charting the seas of information technology. Tech. Rep., 1994.

About the Authors:

PAUL NELSON is an assistant professor at the William E. Simon Graduate School of Business Administration. **Author's Present Address:** William E. Simon Graduate School of Business Administration, University of Rochester, Rochester, NY 14627.

WILLIAM RICHMOND is an associate with Perot Systems Corporation and an associate professor at the Institute of Graduate and Professional Business Studies at George Mason University. **Author's Present Address:** Perot Systems Corporation, 1801 Robert Fulton Drive, Suite 200, Reston, VA. 22091.

ABRAHAM SEIDMANN is the Xerox Professor and Areas Coordinator of Computers and Information Systems, Management Science and Operations Management at the William E. Simon Graduate School of Business Administration at the University of Rochester. **Author's Present Address:** William E. Simon Graduate School of Business Administration, University of Rochester, Rochester, NY 14627; email: SEIDMANNAV@MAIL.SSB.ROCHESTER.EDU.

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage; the copyright notice, the title of the publication, and its date appear; and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© ACM 0002-0782/96/0700 $3.50