Assessing the Economic Benefits of Information Systems Auditing

J. Christopher Westland

School of Business Administration
University of Southern California
Los Angeles, California 90089-1421

As corporate management attempts to extract more end user benefit from information systems department expenditures, interest has grown toward the use of information systems auditing to assure software quality. This research shows that information systems departments are motivated by end users to provide lower quality systems than they would if allowed to pursue their own objectives. The research continues by demonstrating that auditing cannot *a priori* be assumed to raise the quality of corporate information systems. In fact, auditing tends to establish objectives that lower software quality. It demonstrates that audits are most beneficial in managing unsophisticated information systems departments in which end users are currently dissatisfied with their level of support. Augmenting the systems development process via technologies such as computer aided software engineering and prototyping may more consistently and effectively improve quality than does auditing. Recent developments among the large audit firms indicate that they recognize the importance of new software development technology, and are restructuring their businesses accordingly.

Economics of information systems management—EDP auditing—Information strategy and policy—Organizational information processing—System performance assessment

1. Information Systems Auditing

Seven billion dollars are spent annually on external auditing by public accountants in the U.S. It is estimated that a comparable amount is spent on internal auditing. Information systems auditing has grown significantly in the past two decades, due to the exponential growth of computer based information systems which perform corporate financial and administrative functions. The importance of information systems auditing reflects both the high cost of errors in increasingly complex software, and the magnitude of software development expenditures. Schlender (1989) estimated 1989 software expenditures of $112 billion in the U.S. alone.

There is a renewed interest in information systems quality. Correction of systems quality deficiencies may appropriate 75% of the information systems department's budget in large corporations, and can saddle a corporation with a significant financial burden (Swanson 1988). The American Institute of Certified Public Accountants, the Institute of Internal Auditors and the EDP Auditors' Association all have aggressively promoted the conduct of information systems auditing to assure the quality of corporate systems. Yet the effectiveness of auditing expenditures in improving systems quality has received little attention.

1047-7047/90/0103/0309/$01.25

Information systems audits are conducted as part of an external audit by a public accounting firm, during the study and evaluation of internal control as required by the second fieldwork standard of the American Institute of Certified Public Accountants' (AICPA) Generally Accepted Auditing Standards, as presented in their Statements on Auditing Standards (AICPA 1989a). Information systems audits are also conducted by internal corporate auditors. Internal audit reports are often relied upon by the external auditors, and generally support the same objectives as external audits.

Auditing research by Causey (1979), DeJong and Smith (1984), DeJong (1985), and Simunac [1980] has viewed the auditor's responsibility as one of reporting variances below minimum quality standards for information systems operations, controls and so forth. Auditing standards are legislated by various professional organizations, e.g. the AICPA, the Institute of Internal Auditors, and the EDP Auditor's Association. The Statements on Auditing Standards and the AICPA Code of Professional Ethics (AICPA 1989b) establish certain minimum standards for the conduct and interpretation of audit tests. The Rules of Conduct section of the Code of Professional Ethics is enforceable by the AICPA, and may also guide litigation in the courts. Litigation involving the auditor's performance with respect to the Generally Accepted Auditing Standards concerning "due care" and "adequate disclosure" has penalized auditors for failing to include in the auditor's report findings in which corporate performance does not meet minimum quality standards. Additionally the "independence" standard requires that auditors be financially, attitudinally, and politically independent of the organizations and processes which they are auditing. While independence limits professional conflicts of interest, it also leads auditing to focus on situations in which minimum standards have not been met.

Conversely, auditing standards do not address corporate performance which exceeds minimum quality standards. Neither do auditors' reports, letters of recommendation, nor other auditor-client communications. The AICPA's Statements on Auditing Standards repeatedly contend, and auditors have repeatedly argued during litigation, that they are responsible only for finding systems inadequacies which result in "material" errors and omissions. In summary, auditors have publicly argued for limited responsibility extending only to those reporting situations in which systems fall below the minimum standards of quality and result in "material" errors and omissions.

Several previous studies have addressed topics in the area of information systems auditing and information systems quality. McFarlan (1973), Rittenberg and Purdy (1978), Halloran, Manchester, Moriarity, Riley, Rohrman, and Skramstad (1978), Merten and Severance (1981), Ballou and Pazer (1985) and Garvin (1987) addressed auditing. Keidel (1981), Gustafson, and Kerr (1982) and Hansen (1983) considered techniques directed toward attaining minimum quality levels via pre-release testing, and various end user review tasks included in the development cycle. Akerlof (1970) and Leland (1979) investigated market aspects of quality; their analyses provide the basis for the current investigation of information systems auditing. Westland (1989), Kriebel and Raviv (1980), and Milne and Weber (1983) also considered the economics of software quality, and some of this work motivates the current analysis.

The current research develops a short run comparative static analysis of software development which focuses on the quality of software provided to end users. The analysis is short run because the decision making time horizon involved in software development does not allow significant capacity alterations through either hardware reconfiguration or end user department restaffing.

Three agents are assumed in the analysis, each with unique objectives—the information systems department wishes to maximize the value of its services net of the costs of providing them given its fixed capacity; the end users wish to maximize software quality given their expenditures on information systems; and the auditor wishes to alert the end users to software of unacceptable quality. End users are both the purchasers of information systems, and the beneficiaries of auditing.

The fundamental unit of auditing, price, cost and quality measurement is a unit of systems support effort called a "software module." This could conform to measures of support such as delivered source instructions, delivered function points, or other objective units; alternative measures are surveyed in Boehm (1984), Albrecht and Gaffney (1983), and Behrens (1983). Structuring the analysis at this level allows the consideration of both new systems development and maintenance, since the majority of systems maintenance represents the addition of features and reports and may be interpreted as the delivery of new software modules (Swanson 1988).

To facilitate the analysis, this research excludes certain factors which may influence software quality in the long run. The model excludes the cost of auditing itself; only the added costs of software support are included in the model. Thus its conclusions may be conservative. The model also ignores the fact that end users may rely on "signals" from the information systems department to act as their own quality control monitors (Stigler 1961, Spence 1973). Means also exist for improving software quality which burden the end user less than engaging an auditor. For example, management can make the information systems department *ex post facto* responsible for poor software quality by forcing them to absorb all maintenance costs above some agreed upon amount. Quality may also be improved by periodic testing and continuing education. These options lie outside the scope of the current research.

The investigation and analysis evolves as follows. Section 2 describes software supply at a given quality level. Section 3 determines the demand and equilibrium quality level which obtains from typical software development negotiations. Section 4 examines accountant's profit, and determines whether the information systems department has any motivation to improve on the software quality from the equilibrium. Section 5 determines whether auditing motivates the information systems department to improve on equilibrium software quality. Section 6 presents an example which would be suitable for empirical fitting of data, and §7 discusses these results in the context of current industry trends and developments.

2. Software Quality and Software Supply

Quality, as perceived by the end user, may encompass many different facets of the software's life cycle—from compliance of the operating software to the end user's requirements specification, to the physical reliability of the hardware on which it operates. At a minimum, software quality means that the module can be depended upon to produce reliable output supporting the module's ultimate mission. In more sophisticated applications, quality may relate to the "look and feel" of graphics interfaces, elegance of error handling, and other subtle and subjective aspects of good systems development.

Technical measures of quality such as simplicity, legibility, self-description, conciseness, and tolerance are inherent in the software. Software reengineering tools typically generate technical quality measures from a review of existing code, and may compare them to established programming standards through Kiviat diagrams. *Defect* measures, such as number of errors corrected in testing and expected errors based on function points, are derived during verification and validation of software. *End user* measures such as service level and satisfaction are gleaned from end user inquiries after the software has been used for some time. Substandard performance, as measured by end user quality metrics, may motivate the replacement or maintenance of software. Technical and defect measures of quality are often substituted for end user measures, because they are objective, easy to obtain, and may be obtained earlier in the software development life cycle.

Let quality be captured in a single measure θ with finite range $[\theta_1, \theta_2]$. This single measure reflects the fact that the various quality metrics are really surrogates for one factor—the end user perception of quality level. End users perceive information systems as vehicles for accomplishing their tasks, and are only concerned with whether software functions properly in accomplishing these tasks. The finite range of θ reflects the fact that a given metric is valid only over a limited range. Under this definition of θ, most information systems auditing tasks may be classified as vehicles intended to improve the quality of software. The quality level of an arbitrarily selected module may be perceived as a realization θ of the quality random variable $\tilde{\theta}$. The randomness of $\tilde{\theta}$ is an artifact of comparative statics; time is compressed out of the analysis, thus the set of software modules produced over time is unsequenced, and $\tilde{\theta}$ models the distribution of quality over this set of modules.

Let $g(\theta)$ be the potential supply of software modules which the information systems department can produce at quality level θ during a period. In practice, $g(\theta)$ will be some positive integer, but without loss of generality may be depicted as a continuous measure. Visualize the nature of supply by supposing that the best programmer or software development team in a department can only produce a limited number of modules $g(\hat{\theta})$ per period, and this will all be done at the highest quality level $\hat{\theta}$; the next best programmer can produce $g(\hat{\hat{\theta}})$ at some quality level $\hat{\hat{\theta}} < \hat{\theta}$; and so forth for the remainder of the departmental production capacity. The result is the supply $g(\theta)$ or frequency of occurrence of modules with quality θ for a quality random variable $\tilde{\theta}$ defined on the set of software modules. $G(\theta) = \int_{\theta_1}^{\theta} g(t)\,dt$ is the information systems department's total production capacity for software modules of quality less than θ. The total departmental capacity, and thus potential supply of software modules in a period, is $G(\theta_2) = \int_{\theta_1}^{\theta_2} g(t)\,dt$.

The production of a certain quantity of modules at quality level θ reflects capacity constraints resulting from different information systems department's employees producing at different levels of quality. In practice, the quality level of an employee's output is dependent upon education, experience, personal traits, job satisfaction and so forth. In the long run, this quality level is also dependent on the job market for information systems personnel, and may be difficult to influence internally, particularly when employee turnover is high. Thus, information systems production capacity is fixed in the short run.

Define a normalized measure of quality $\tilde{\xi} = G(\tilde{\theta})/G(\theta_2) \in [0, 1]$. The transformation $G(\tilde{\theta})/G(\theta_2)$ simply selects a different measurement system for quality; but one with the desirable mathematical property that the supply function $f(\xi)$ of $\tilde{\xi}$ is a uniform probability density function on [0, 1]. Monte Carlo simulations use the inverse of this transformation to convert a uniform random number generator to a generalized random variable generator, and Leland (1979) uses a similar device to extend Akerlof's (1970) analysis; thus the transformation is not novel. Since normalized departmental capacity is $F(1) = \int_0^1 f(t)dt = 1$, the probability density function $f(\xi)$ may be interpreted as the proportion of software module production capacity supplied at normalized quality measure ξ. Since the transformation is isomorphic and $G(\theta)$, θ_1 and θ_2 are known, $\tilde{\theta}$ may be recovered at any time from $\tilde{\xi}$. Using this transformation allows the following analysis to be valid for any supply function $g(\theta)$.

Denote the corresponding marginal cost of supplying a module of quality ξ as $\kappa(\xi)$. This is the opportunity cost of providing an additional software module at a quality level which is as high or higher than any of the previously produced software modules. The information systems department can provide software modules with quality $\xi \leq \hat{\xi} \in [0, 1]$ where $\hat{\xi}$ is the maximal quality level at which the information systems department is able to produce. Maximal quality $\hat{\xi}$ will vary among information systems departments, due to differences in local labor markets for programmers and analysts, differing management styles, and differing complexities of the computing environment.

Quality metrics such as number of defects corrected in testing are lower at higher quality levels; other measures directly correlate with quality. The transformation of actual quality measurements $\tilde{\theta}$ into a normalized quality $\tilde{\xi} \in [0, \hat{\xi}]$ provides a random variable with a uniform distribution for which marginal cost κ increases with increasing quality ξ

$$\partial\kappa(\xi)/\partial\xi > 0. \tag{1}$$

Increasing marginal cost occurs at the high end of the quality range θ_2 because an organization must make ever increasing investments in order to extract a little extra quality from the software development process. Brooks (1975), for example, argues that complex interdependence of software components causes the marginal cost of implementing new features and controls in systems to rise faster than the additional quality rendered. Costs of software development may roughly be dichotomized into labor, e.g. programming and analysis; and capital, e.g. computer use, cost of computer aided software engineering (CASE), and equipment purchases. Generally, higher pay should procure better programmers and analysts, and thus lead to higher quality systems. Higher capital investments, e.g. dedicated CASE workstations for each analyst or programmer, should make it easier to keep track of complex requirements specifications and designs, and should ease the task of debugging. This in turn should lead to higher quality systems. But there must be some practical limit to the level of spending on software modules, and thus some maximum quality level $\hat{\xi}$ is anticipated. Quality control costs can be high. For example, Lotus 1-2-3 version 3 contains 400,000 lines of code, with quality control and testing costs totalling \$15 million—over twice the \$7 million cost of development (Schlender 1989).

At the low end of quality θ_1, the cost-quality relationship $\partial\kappa(\xi)/\partial\xi > 0$ may break down due to excessive error correction and rewriting of programs, employee morale and turnover problems, and ineffective equipment use and procurement. Kull (1986) related an extreme case in which defects became so numerous that error correction caused enormous cost overruns. This extreme reflects an abnormal and uncontrolled software development process which is outside the scope of the current analysis, thus the lower bound on quality $\tilde{\theta}$.

3. Software Demand and Equilibrium Quality

This section identifies the equilibrium level of software quality resulting from end user demand for services and the previous supply considerations. The end user and the information systems department contract for delivery of a software module, or for a set of modules comprising an information system, based on price, cost and quality of software to be delivered. There exists an information asymmetry—the end user does not know the quality of software that the information systems department will ultimately deliver. The end user can evaluate quality only after using the software for an extended period of time, and thus lacks quality information at the time of contracting.

Assume that end user demand, or willingness to pay for software $\delta(\rho, \eta)$ is dependent upon some measure ρ of the information systems department's "reputation" for quality; and on the quantity of delivered software modules η. In the normalized quality measurement system defined in the prior section, ρ and η are measured in the same units as ξ, thus quantity η is actually the proportion of total software module capacity that can be delivered, rather than an integer count of software modules.

Inefficiencies arising from information asymmetry in this analysis are due to the end users adopting a constant reputation value ρ when actual quality ξ varies. This characterization of information asymmetry is consistent with prior literature in the economics of imperfect information. For example, Stigler (1961) defines an "unknown true state of the world" and a "signal." The information structure in his setup is asymmetric because information is complete on the selling side; while incomplete information characterizes the buying side. A "signal," e.g. ρ, tells the buying side something about the "true state," e.g. ξ. Milgrom and Roberts (1987) describe information asymmetries which arise when one agent in a contract possesses more information about the subject of the contract, in this case the quality ξ of the next software module, than another agent. They also assume that there exists some publicly available information, such as reputation ρ, for the problem to be interesting.

The end user's "signal" for future software quality is the information systems department's reputation based on the quality of software delivered in the past. Reputation has been discussed in Becker (1957, 1973, 1974), Arrow (1972) and Akerlof (1980). Akerlof summarizes the reputation definitions used by Becker, Arrow and himself as a linear function of two parameters—an individual's obedience or disobedience to a community's code of behavior, and the proportion of the population who believe in the code. Akerlof's reputation statistic is consistent with the expectation of a random variable representing "obedience/disobedience" outcomes. Adapting Akerlof's definition to the model presented in this research, if all of the end users believe that quality is important, then $\tilde{\xi}$ may be perceived as a random variable of "obedience/disobedience" outcomes, and $\rho = \bar{\xi}$, the arthimetic mean of prior quality levels of delivered software, is an appropriate linear reputation statistic.

Given the information systems department's reputation $\bar{\xi}$, end users will only expect to receive, and will only be willing to pay for a module with quality $\bar{\xi}$. Thus the end user willingness to pay for services $\delta(\bar{\xi}, \eta)$ is dependent upon the quantity of software η to be delivered and the average quality $\bar{\xi}$ expected for modules produced by the information systems department. The quantity of software modules that the information systems department will potentially supply when maximal quality is $\hat{\xi}$ is

$$\eta = \int_0^{\hat{\xi}} dF(t) = \hat{\xi} \tag{2}$$

and $\rho = \bar{\xi} = \hat{\xi}/2$ since $f(\xi)$ is uniform on $[0, \hat{\xi}]$. Thus the willingness of the end users to pay given maximal quality of software $\hat{\xi}$ is

$$\delta(\rho, \eta) = \delta(\hat{\xi}/2, \hat{\xi}). \tag{3}$$

End users will be willing to pay more for software modules produced by an information systems department with a reputation for quality, i.e., $\partial\delta(\rho, \eta)/\partial\rho > 0$. Additionally, end users will satisfy their most pressing software needs first, and thus demand satiates, $\partial\delta(\rho, \eta)/\partial\eta < 0$. There is empirical and anecdotal evidence to support the saturation of end user needs; e.g. in Nolan's (1979) stage hypothesis toward the end of stage two, end users tend to have most of their pressing systems needs fulfilled, and start to concern themselves with control of costs.

An equilibrium obtains when both parties have knowledge of each others preferences, $\delta(\rho, \eta)$ and $\kappa(\xi)$. These preferences will have been communicated during the feasibility study, requirements specification, design and release negotiations for the specific information system containing the software modules. At the equilibrium maximal quality $\hat{\xi}^*$, the information systems department's willingness to supply and the end user's willingness to fund are equal, and both parties agree to the software development. The equilibrium with imperfect information satisfies

$$\delta(\hat{\xi}^*/2, \hat{\xi}^*) = \kappa(\hat{\xi}^*) \tag{4}$$

where $\delta(\hat{\xi}^*/2, \hat{\xi}^*)$ and $\kappa(\hat{\xi}^*)$ are the respective price and opportunity cost of producing an additional software module at the quality margin. This equilibrium reflects the outcome of a series of "bids" and "asks" embodied in the software development negotiation process. Results from welfare economics, e.g. Olson (1965), suggest that this is not optimal; only that $\hat{\xi}^*$ is the maximal quality level that will cause willingness to pay and willingness to provide to coincide. In fact, $\hat{\xi}^*$ is likely to be suboptimal because it results from negotiations with asymmetric information—at the quality margin the end users think they will get a software module of quality $\bar{\xi}$, and the information systems department thinks it will provide a software module of quality $\hat{\xi}$. Thus the end users will always try to force down the equilibrium price in negotiations, possibly to their own detriment by lowering the general quality of software delivered.

Both $\partial\delta(\rho, \eta)/\partial\rho$ and $\partial\kappa(\hat{\xi})/\partial\hat{\xi}$ may be greater than zero by the previous arguments. Thus under certain situations, both willingness to pay $\delta(\rho, \eta)$ and willingness to supply $\kappa(\hat{\xi})$ may be increasing in $\hat{\xi}$. Demand price $\delta(\rho, \eta)$ must be greater than marginal cost $\kappa(\hat{\xi})$ to the left of the equilibrium $\hat{\xi}^*$, and less to the right. Otherwise, at the margin the end user will not be willing to pay for the next higher quality level module, no matter what is the current level of quality, and $\hat{\xi}^* = 0$ with consequent zero quantity production $\eta = 0$. Stated otherwise, unless $\kappa(\hat{\xi})$ is increasing faster than $\delta(\rho, \eta)$ around $\hat{\xi}^*$, there will not be a meaningful equilibrium. The equilibrium $\hat{\xi}^*$ requires the additional constraint that

$$\frac{1}{2}\left.\frac{\partial\delta(\rho, \hat{\xi})}{\partial\rho}\right|_{\rho=\hat{\xi}/2} + \left.\frac{\partial\delta(\hat{\xi}/2, \eta)}{\partial\eta}\right|_{\eta=\hat{\xi}} < \frac{\partial\kappa(\hat{\xi})}{\partial\hat{\xi}} \tag{5}$$

in the neighborhood $\hat{\xi} = \hat{\xi}^*$.

4. Software Quality Without Auditing

The prior section determined the equilibrium quality that will obtain from typical software development negotiations. The current section investigates whether the information systems department, in the absence of external forces such as auditing, is motivated to produce at higher levels of quality than the equilibrium. Software production at higher quality levels is presumed to be beneficial to the end users and to the firm.

The information systems department makes choices concerning software quality and quantity based on some objective. Spence (1975) and Mendelson (1985) proposed a *net value* measure as the objective of such choices. The net value Π of software is

$$\Pi = \int_0^{\eta} \delta(\rho, t)\,dt - \int_0^{\hat{\xi}} \kappa(t)\,dt \tag{6}$$

for some quantity η and maximal quality $\hat{\xi}$. This objective function is adopted for the current analysis.

Consider the situation in which the end users have *perfect information* about quality, and are able to make decisions based on a *variable* $\rho = \xi$ that is precisely the quality level at which the information systems department produces the next software module. The equilibrium obtained in equation (4) changes under perfect information; equilibrium obtains for some $\hat{\xi}^0$ such that

$$\delta(\hat{\xi}^0, \hat{\xi}^0) = \kappa(\hat{\xi}^0). \tag{7}$$

This equilibrium $\hat{\xi}^0$ must be optimal since the contract is made with perfect information on both sides. With perfect information and $\eta = \hat{\xi}$ from equation (2)

$$\begin{aligned} \left.\frac{\partial \Pi}{\partial \hat{\xi}}\right|_{\hat{\xi}=\hat{\xi}^0} &= \frac{\partial}{\partial \hat{\xi}}\left[\int_0^{\hat{\xi}}\left[\int_0^{\hat{\xi}} \delta(t, s)\,ds - \kappa(t)\right]dt\right]_{\hat{\xi}=\hat{\xi}^0} \\ &= [\delta(\hat{\xi}, \hat{\xi}) - \kappa(\hat{\xi})]|_{\hat{\xi}=\hat{\xi}^0} = \delta(\hat{\xi}^0, \hat{\xi}^0) - \kappa(\hat{\xi}^0) \equiv 0. \end{aligned} \tag{8}$$

So perfect information eliminates the information asymmetry, and net value is maximized at the equilibrium $\hat{\xi}^0$ in equation (7).

With *imperfect information*, objective function Π motivates the information systems department to provide its optimal level of quality *if* at the equilibrium quality level $\hat{\xi}^*$ obtained in equation (4), the derivative of Π with respect of $\hat{\xi}$ at $\hat{\xi}^*$ is zero. If this derivative is greater than zero, then net value is increasing with higher levels of quality at the margin, and the information systems department is underproviding quality to its own detriment, and to the detriment of end users and the firm. If this derivative is less than zero, then net value is decreasing with higher levels of quality at the margin, and the information systems department is overproviding quality to its own detriment. Substituting $\hat{\xi} = \eta$ and $\hat{\xi}/2 = \bar{\xi}$ from the prior section, and differentiating the definite integral with respect to $\hat{\xi}$ via Leibniz's Theorem gives

$$\frac{\partial \Pi}{\partial \hat{\xi}} = \int_0^{\hat{\xi}} \frac{1}{2}\left[\left.\frac{\partial \delta(\rho, \eta)}{\partial \rho}\right|_{\rho=\hat{\xi}/2}\right]d\eta + \delta(\hat{\xi}/2, \hat{\xi}) - \kappa(\hat{\xi}). \tag{9}$$

At $\hat{\xi} = \hat{\xi}^*$ the rightmost two terms in equation (9) net to zero as a result of the equilibrium condition from equation (4), and the derivative of net value evaluated at $\hat{\xi}^*$ is

$$\left.\frac{\partial \Pi}{\partial \hat{\xi}}\right|_{\hat{\xi}=\hat{\xi}^*} = \int_0^{\hat{\xi}^*} \frac{1}{2}\left[\left.\frac{\partial \delta(\rho, \eta)}{\partial \rho}\right|_{\rho=\hat{\xi}^*/2}\right]d\eta > 0. \tag{10}$$

Thus the information systems net value is still increasing around the equilibrium $\hat{\xi}^*$, and could be increased further if the information systems department were willing to produce software modules at a higher maximal level of quality. Equation (10) implies an incentive problem, since the value of $\hat{\xi}$ that maximizes Π always lies at a value higher than the equilibrium value $\hat{\xi}^*$ which obtains from negotiations. The information systems department, in the absence of external motivation to improve quality, will underprovide quality. This concurs with intuition, since at the quality margin the information systems department must deliver a software module priced as if it has average quality $\bar{\xi}$, but it must produce it at a cost associated with highest quality $\hat{\xi}$. Thus the department is underremunerated for the quality of work that it performs, and is undermotivated to develop high quality software modules.

In practice *accounting profit P* rather than net value Π determines the information systems department's performance. Information systems departments may nominally operate as "free" services, cost centers or as profit centers. Allen (1987) argued that the profit center approach provides more efficient and effective information systems service, and increases end user satisfaction. But Mendelson (1985) found that both profit and cost center approaches create inefficiencies which detrimentally affect end user satisfaction. Since, external audits set as an objective the attestation of fairness of reported net income, i.e. profit, and internal audits support the objectives of external audits, there is significant motivation for an information systems department to act as a profit center. Thus accounting profit is the typical objective function used in practice. Optimizing on accounting profit may not be consistent with net value maximization.

End users procure software from within the firm, and the information systems department holds a monopoly position in software. The information systems department's accounting profit P for some delivered quantity of software η is comprised of the total revenues from providing software modules to the end users, less the total cost incurred in supplying those modules (Varian 1984). Thus

$$P = \int_0^{\hat{\xi}} \left[\int_0^{\eta} \delta(\rho, s) \left[1 + \frac{1}{\epsilon(s)} \right] ds - \kappa(t) \right] dt \tag{11}$$

where the price elasticity of quantity demanded is

$$\epsilon(\eta) = \frac{\delta(\rho, \eta)}{\eta} \frac{\partial \eta}{\partial \delta(\rho, \eta)} \leq 0. \tag{12}$$

Assume infinite elasticity, $|\epsilon(\eta)| = \infty$; then $[1 + 1/\epsilon(\eta)] = 1$ and accounting profit coincides with net value.

But the previous assumption that $\partial\delta(\rho, \eta)/\partial\eta < 0$ is inconsistent with infinite elasticity, and assures that $|\epsilon(\eta)| < \infty$. With finite elasticity, $[1 + 1/\epsilon(\eta)] < 1$ for any quantity of software production η, and thus $\partial P/\partial\hat{\xi}|_{\hat{\xi}=\hat{\xi}^0} < 0$. Profit is downward sloping at $\hat{\xi}^0$ and is maximized for some value less than the optimal equilibrium value $\hat{\xi}^0$. This reduction in profit due to finite elasticity $|\epsilon(\eta)| < \infty$ results from saturation of end user computing needs as the quantity η of delivered software increases. The profit maximizing information systems department will choose to produce at some value less than $\hat{\xi}^0$. The elasticity effect on accounting profit is related to *quantity* of production rather than *quality* of production, and tends to obscure the influence of software quality on the end users demand. Net value, on the other hand, captures the perceived value of the software developed, as assessed by the end users, less the cost of production; it ignores demand satiation.

In general, the maximization of accounting profit motivates the information systems department to supply software of suboptimal quality, and appears to provide an unsatisfactory objective function. At the same time, the attestation of accounting profit by auditors, in a sense, forces the adoption of accounting profit as the *de facto* objective function for information systems department activity.

Underprovision of quality in systems development is commonly observed, and is reflected in the high level of post-delivery corrective maintenance identified by Lientz and Swanson (1980) and others. Low software quality is also a major motivation for conducting information systems audits in corporations. The next section investigates the effectiveness of information systems auditing in improving systems quality.

5. Software Quality Where Audits are Performed

Low software quality has been a persistent concern of end users, who have sought various approaches to rectify the problem. One approach has been the conduct of audits which enforce some minimum software quality standard $\xi_0 > 0$ by alerting the end users to quality deficiencies. The minimum quality standard ξ_0 may either be assumed to be enforced perfectly by audits or, without loss of generality, may be viewed as a target or expected minimum quality standard. End users commonly presume that auditing improves the expected quality of delivered software modules. The current section investigates whether the information systems department, under the influence of auditing, is motivated to produce at higher levels of quality than the equilibrium. It investigates the behavior of Π as the minimum quality standard moves upward from $\xi_0 = 0$, i.e. where audits are not performed, to $\xi_0 > 0$, i.e., where audits are performed.

With auditing, information systems support is provided at quality level $\xi \in [\xi_0, \hat{\xi}]$, and low quality support $\xi < \xi_0$ is eliminated. But the expected quality of support $\bar{\xi}$ will also be raised causing software support costs $\kappa(\xi)$ to rise. The net value of auditing is not readily apparent. With auditing, the proportion of capacity provided by the information systems department becomes

$$\eta = \int_{\xi_0}^{\hat{\xi}} dF(t) = \hat{\xi} - \xi_0 \tag{13}$$

and expected quality level is the expectation of a uniform random variable on $[\xi_0, \hat{\xi}]$

$$\rho = \bar{\xi} = \frac{\hat{\xi} + \xi_0}{2}. \tag{14}$$

Substituting these two values gives an equilibrium level of quality $\hat{\xi}^*$ where

$$\delta\left(\frac{\hat{\xi}^* + \xi_0}{2}, \hat{\xi}^* - \xi_0\right) = \kappa(\hat{\xi}^*). \tag{15}$$

Again note that both $\partial\delta/\partial\hat{\xi}$ and $\partial\kappa/\partial\hat{\xi}$ may be greater than zero. Thus, unless κ is increasing faster than δ around $\hat{\xi}^*$ there will not be a meaningful equilibrium, and the additional constraint

$$\frac{1}{2}\left.\frac{\partial\delta(\rho, \hat{\xi} - \xi_0)}{\partial\rho}\right|_{\rho=[\hat{\xi}+\xi_0]/2} + \left.\frac{\partial\delta(\hat{\xi}/2, \eta)}{\partial\eta}\right|_{\eta=\hat{\xi}-\xi_0} < \frac{\partial\kappa(\hat{\xi})}{\partial\hat{\xi}} \tag{16}$$

around $\hat{\xi}^*$ is required. Net value Π is

$$\Pi = \int_0^{\eta} \delta(\rho, t)\,dt - \int_{\xi_0}^{\hat{\xi}} \kappa(t)\,dt \tag{17}$$

where $\eta = \hat{\xi} - \xi_0$ and $\rho = \hat{\xi} - \xi_0/2$ from equations (13) and (14) respectively.

If $\partial\Pi/\partial\xi_0$ is still increasing at $\xi_0 = 0$, improvement in the information systems department's net values may be achieved through information systems auditing which facilitates higher (positive) values of ξ_0. If net value Π can be increased by auditing, then through the arguments presented in the prior section, the information systems department will be motivated to provide support at higher quality levels. Differentiate, combine equilibrium condition (15), and simplify terms, to get an equilibrium $\hat{\xi}^*$

$$\frac{\partial\Pi}{\partial\xi_0} = \frac{1}{2}\left[\int_0^{\hat{\xi}^*-\xi_0} \left.\frac{\partial\delta(\rho,\eta)}{\partial\rho}\right|_{\rho=[\hat{\xi}^*+\xi_0]/2} d\eta\right]\left[\frac{\partial\hat{\xi}^*}{\partial\xi_0}+1\right] - \delta\left(\frac{\hat{\xi}^*+\xi_0}{2}, \hat{\xi}^*-\xi_0\right) - \kappa(\hat{\xi}^*)\frac{\partial\hat{\xi}^*}{\partial\xi_0} + \kappa(\xi_0). \quad (18)$$

Note that $\partial\hat{\xi}^*/\partial\xi_0$ is non-zero because $\hat{\xi}^*$ depends upon δ and κ which in turn depend upon ξ_0. Equation (18) may be positive or negative around $\xi_0 = 0$ given various parameter values. Thus there are circumstances under which audits will cause the information systems department to underprovide software quality, and where it may be undesirable to enforce minimum quality standards via information systems auditing. These circumstances are investigated in the subsequent example.

6. An Example

Consider the following example which elucidates the impact of auditing on systems quality. Commonly used regression and variance analysis statistics assume linear causal models of real world behavior, thus the example can support confirmatory analysis by providing a model for fitting data. Let end user demand depend linearly upon reputation and total quantity of software modules

$$\delta(\rho,\eta) = \frac{1}{2} + \alpha\rho - \beta\eta > 0. \quad (19)$$

In this case, the end user is willing to pay $\frac{1}{2}$ for the first software module produced. Willingness to pay for subsequent modules depends upon reputation ρ less the satiation of end user information processing needs resulting from each implemented system, as reflected in quantity η. Commensurate with inequality (1), define a simple marginal cost function which increases faster than maximal quality

$$\kappa(\hat{\xi}) = \hat{\xi}^2. \quad (20)$$

Figure 1 shows demand and supply of quality for values of $\alpha = \beta = 1$. Auditing encourages the information systems department to improve on the equilibrium quality level when net values are increasing in ξ_0 at the point where $\xi_0 = 0$. This is the point of movement from no auditing, to auditing which enforces some positive quality standard $\xi_0 > 0$. In the example, net values are increasing in ξ_0 where

$$\left.\frac{\partial\Pi}{\partial\xi_0}\right|_{\xi_0=0} = \frac{[\alpha\hat{\xi}^* - [\hat{\xi}^*]^2]\left[\frac{\alpha}{2}+\beta\right]}{\sqrt{\left[\frac{\alpha}{2}-\beta\right]^2+2}} - \frac{1}{2} + \left[\frac{3\alpha}{2}-\beta\right]\hat{\xi}^* > 0. \quad (21)$$

This result is derived in the Mathematical Appendix, and figure 2 shows its graph. For the majority of values of α and β plotted in figure 2, it appears that raising ξ_0 lowers net value Π—information systems auditing appears to be counterproductive

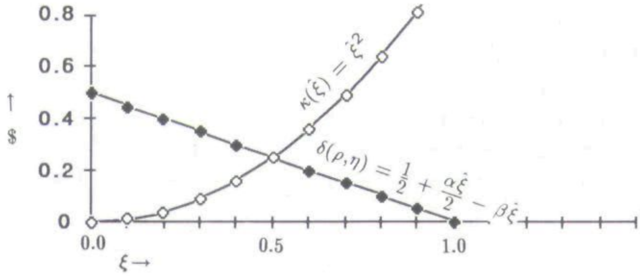

FIGURE 1. Demand Price $\delta(\rho, \eta)$ and Marginal Cost $\kappa(\hat{\xi})$.

in motivating the information systems department to provide higher qualify software. Information systems auditing appears to be most effective in improving quality when the following conditions exist.

(1) Where α is large, provided the denominator of $\partial\Pi/\partial\xi_0$ remains positive. Large α implies that the information systems department's reputation is important to end users, and that they are willing to pay more for a given amount of improvement in reputation, i.e. average software quality.

(2) Where β is small. Small β implies that end users are more concerned with quality than with quantity of production.

(3) Where $\partial\kappa(\xi)/\partial\xi|_{\xi=\hat{\xi}^*} = 2\hat{\xi}^*$ is small. When the maximum quality level at equilibrium is small, then the marginal cost of improving quality is small for the amount of improvement achieved.

Condition (1) will exist where end users are currently dissatisfied with the level of service they are receiving. Condition (2) is expected to exist in most corporations for two reasons—information systems budgets are often "locked in" by prior commitments, staff sizes and so forth, and most corporations have an installed base of transaction processing systems which meet their most pressing information systems needs. Condition (3) is more difficult to assess. Quality may be improved by the provision of software modules with improved controls and error handling capability, better hu-

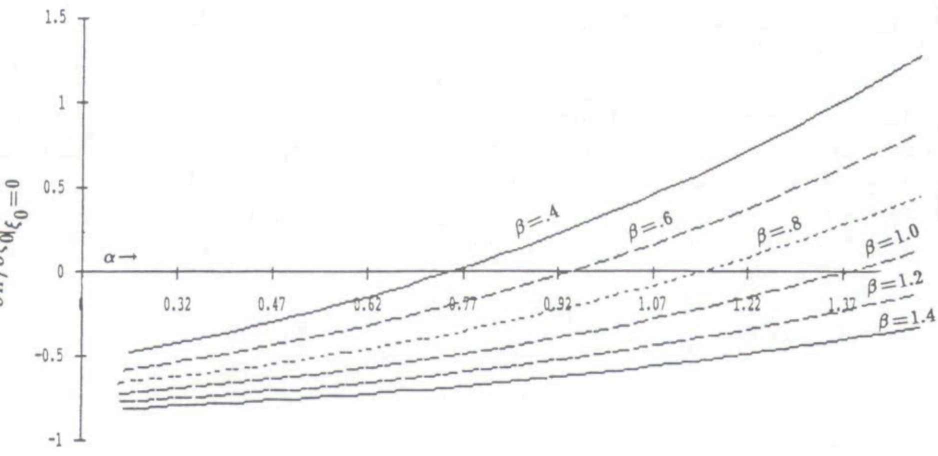

FIGURE 2. The Behavior of $\left.\frac{\partial\Pi}{\partial\xi_0}\right|_{\xi_0=0}$.

man interfaces, and so forth; but all of these add cost and complexity at an increasing rate. Condition (3) will tend to exist in developing information systems departments, e.g. those experiencing Nolan's (1979) development stages one to three, where systems are simple and not highly interdependent. The major insight from this example, then, is that information systems auditing may be most beneficial where information systems are relatively unsophisticated, and where end users are currently dissatisfied with the level of software quality they are receiving. Since the example uses a normalized quality measure ξ, these conclusions should be robust for moderate perturbations of the demand and supply curves in the example, and thus may be valid in a wide range of real world environments.

7. Discussion

The model presented in this research shows that auditing cannot be assumed *a priori* to improve the quality of corporate information systems. Rather, the benefits from auditing tend to be situation specific. Indeed, audit attestation of accounting profit may motivate information systems departments to underprovide quality.

This result is not surprising if one considers the results of imposing minimal quality standards in other disciplines, particularly production. Over forty years ago, Deming (1950) presented evidence that grafting quality control inspections onto existing production processes failed to consistently improve quality. Deming contended that fundamental changes to production processes were required to improve production quality. In the current analysis, it is evident that audits which enforce a lower quality bound ξ_0 do not necessarily improve systems quality. To improve systems quality, it is necessary to improve the overall quality of production, i.e. to raise $\hat{\xi}$ to some higher value.

If information systems auditing and similar approaches to assuring minimal quality levels in software are counterproductive, what alternative approaches are available to improve software quality? Insight is provided by recent trends and fundamental changes in information systems development which promise to increase the quality of software production.

One such development involves risk-managed "spiral" or rapid prototyping approaches to systems development, which seek to control quality and cost of a system in initially small, but expanding, sets of tasks (Boehm 1986, 1984, Westland 1990a). After each prototype, end users evaluate the actual operation of a prototype and one of three decisions is made—go on to build the next prototype; revise the system's design and then build the next prototype; or discontinue development. To place this in the context of the previous analysis, the feedback from each prototype serves to eliminate a part of the information asymmetry which tends to force down the equilibrium $\hat{\xi}^*$ in equation (4). So rather than reputation $\bar{\xi}$ driving the willingness of end users to pay down to $\delta(\bar{\xi}, \eta)$, the transfer of information during prototype evaluation will place the end user's willingness to pay at a value closer to that provided with perfect information $\delta(\hat{\xi}, \eta)$. In this situation, the information systems department is motivated to produce at its net value maximizing levels of quality, and not underprovide quality. Similarly, quality circles and other methods of fostering communication between end users and the information systems department also move end user willingness to pay from $\delta(\bar{\xi}, \eta)$ to $\delta(\hat{\xi}, \eta)$.

Rapid prototyping has become feasible on a wide scale since the introduction of various CASE tools which incorporate graphic design support, code generation, validation, reengineering, and hypertext based documentation (Westland 1990a, 1990b, 1990c). These tools are generally acquired externally by corporations, with the large audit firms having become major suppliers. The evolution of new systems development technologies reflects the end users' demand for improved software quality. Strategic necessity will probably force firms in the future to invest in CASE, just as today's firms are required to invest in compilers and operating systems.

Another fundamental change in systems development philosophy is reflected in the current worldwide restructuring of the major audit firms. The major audit firms have increasingly emphasized information systems consulting rather than auditing. Some of this restructuring has been required by the automation of most corporate accounting and finance functions. The clients of the major audit firms are thus end users of many corporate information systems. But the major audit firms may also recognize that better systems development technologies rather than better auditing, present the most effective way to improve systems quality. As automated information systems become complex and ubiquitous fixtures on the corporate landscape, audit clients are pulling the major audit firms into information systems consulting and away from traditional auditing. The major audit firms appear to be adapting to this demand through their growing emphasis on software consulting and automated development technologies such as CASE. The results of this research suggest that their response is natural and salutary, and can be expected to accelerate in the future.*

* Haim Mendelson, Associate Editor. This paper was received on August 6, 1988, and has been with the authors $3\frac{1}{2}$ months for 4 revisions.

Mathematical Appendix

In the example, end user demand depends upon average quality and total quantity of software modules

$$\delta(\rho, \eta) = \frac{1}{2} + \alpha\rho - \beta\eta. \tag{22}$$

Marginal cost is

$$\kappa(\hat{\xi}) = \hat{\xi}^2. \tag{23}$$

Combining with (14), (15) and substituting these into (16) gives

$$\begin{aligned}\kappa(\hat{\xi}^*) - \delta\left(\frac{\hat{\xi}^* + \xi_0}{2}, \hat{\xi}^* - \xi_0\right) &= [\hat{\xi}^*]^2 - \alpha\frac{\hat{\xi}^* + \xi_0}{2} + \beta[\hat{\xi}^* - \xi_0] - \frac{1}{2}\\ &= [\hat{\xi}^*]^2 - \left[\frac{\alpha}{2} - \beta\right]\hat{\xi}^* - \left[\frac{\alpha}{2} - \beta\right]\xi_0 - \frac{1}{2} = 0\end{aligned} \tag{24}$$

at equilibrium. Solve the quadratic equation to get

$$\hat{\xi}^* = \frac{\left[\frac{\alpha}{2} - \beta\right] + \sqrt{\left[\frac{\alpha}{2} - \beta\right]^2 + 4\left[\frac{\alpha}{2} + \beta\right]\xi_0 + 2}}{2} \tag{25}$$

with

$$\frac{\partial\hat{\xi}^*}{\partial\xi_0} = \frac{\left[\frac{\alpha}{2} + \beta\right]}{\sqrt{\left[\frac{\alpha}{2} - \beta\right]^2 + 4\left[\frac{\alpha}{2} + \beta\right]\xi_0 + 2}}. \tag{26}$$

The rate of change in net value Π with the minimum quality level ξ_0 enforced by auditing is

$$\frac{\partial \Pi}{\partial \xi_0} = \alpha[\hat{\xi}^* - \xi_0]\left[\frac{\left[\frac{\alpha}{2}+\beta\right]}{\sqrt{\left[\frac{\alpha}{2}-\beta\right]^2 + 4\left[\frac{\alpha}{2}+\beta\right]\xi_0 + 2}} + 1\right] - \frac{1}{2} + \alpha\,\frac{\hat{\xi}^* + \xi_0}{2} - \beta[\hat{\xi}^* - \xi_0] - [\hat{\xi}^*]^2\left[\frac{\left[\frac{\alpha}{2}+\beta\right]}{\sqrt{\left[\frac{\alpha}{2}-\beta\right]^2 + 4\left[\frac{\alpha}{2}+\beta\right]\xi_0 + 2}}\right] + \xi_0^2 \quad (27)$$

and thus

$$\left.\frac{\partial \Pi}{\partial \xi_0}\right|_{\xi_0=0} = \frac{\left[\alpha\hat{\xi}^* - \left[\hat{\xi}^*\right]^2\right]\left[\frac{\alpha}{2}+\beta\right]}{\sqrt{\left[\frac{\alpha}{2}-\beta\right]^2 + 2}} - \frac{1}{2} + \left[\frac{3\alpha}{2} - \beta\right]\hat{\xi}^* \quad (28)$$

in the neighborhood of $\xi_0 = 0$ where $\hat{\xi}^*$ is given by (25). Around the equilibrium quality level $\bar{\xi}^*$, $\delta \approx \kappa$, and $\xi \in [0, 1]$ implies $\bar{\xi} \in [0, \frac{1}{2}]$. Thus ranges of $\alpha, \beta \in [0, \frac{3}{2}]$ seem reasonable, and the graph in figure 2 depicts the behavior of $\partial\Pi/\partial\xi_0|_{\xi_0=0}$ over these values.

References

Albrecht, A. J. and J. E. Gaffney, "Software Function, Source Lines of Code and Development Effort Prediction," *IEEE Transactions on Software Engineering,* SE-9,6 (November 1983), 639–647.

Akerlof, G. A., "The Market for Lemons," *Quart. J. Economics,* 84 (1970), 488–500.

Akerlof, G. A., "The Theory of Social Custom, of Which Unemployment May Be One Consequence," *Quart. J. Economics,* 94 (1980), 749–775.

Allen, B., "Make Information Services Pay its Way," *Harvard Bus. Rev.,* (Jan–Feb 1987), 57–63.

American Institute of Certified Public Accountants, *Statements on Auditing Standards,* AICPA, New York, NY, 1989a.

American Institute of Certified Public Accountants, *Code of Professional Ethics,* AICPA, New York, NY, 1989b.

Arrow, K. J., "Models of Job Discrimination," in A.H. Pascal (ed.), *Racial Discrimination in Economic Life,* Heath, Lexington, MA (1972), 83–102.

Ballou, D. P. and H. L. Pazer, "Modeling Data and Process Quality in Multi-Input, Multi-Output Information Systems," *Management Science,* 31, 2 (1985), 150–162.

Becker, G. S., *The Economics of Discrimination,* University of Chicago Press, Chicago, IL, 1957.

Becker, G. S., "A Theory of Marriage: Part I," *J. of Political Economy,* 81 (1973), 813–846.

Becker, G. S., "A Theory of Marriage: Part II," *J. of Political Economy,* 82 (1973), 11–26.

Behrens, C. A., "Measuring the Productivity of Computer Systems Development Activities with Function Points," *IEEE Trans. Software Engineering,* SE-9,6 (November 1983), 648–651.

Boehm, B. W., "Software Engineering Economics," *IEEE Trans. Software Engineering,* SE-10 (January 1984), 1–43.

Boehm, B. W., "A Spiral Model of Software Development & Enhancement," *ACM SIGSOFT Engineering Notes,* 11, 4 (August 1986), 21–42.

Boger, D. C. and N. R. Lyons, "The Organization of the Software Quality Assurance Process," *Data Base,* 16, 2 (1985), 11–15.

Brooks, Jr. F. P., *The Mythical Man-Month,* Addison-Wesley, Reading, MA, 1975.

Causey, D. H. *Duties and Liabilities of Public Accountants,* Dow Jones-Irwin, Homewood IL, 1979.

DeJong, D. V. and J. H. Smith, "The Determination of Audit Responsibilities: An Application of Agency Theory," *Auditing: A Journal of Theory and Practice,* 4 (1984), 20–34.

DeJong, D. V., "Class-Action Privileges and Contingent Legal Fees: Investor and Lawyer Incentives to Litigate and the Effect on Audit Quality," *J. Accounting and Public Policy,* 4 (1985), 175–200.

Deming, E., *Some Theory of Sampling,* Wiley, New York, NY, 1950.

Garvin, D. A., "Competing on the Eight Dimensions of Quality," *Harvard Bus. Rev.,* 65, 6 (1987), 101–109.

Gustafson, G. G. and R. J. Kerr, "Some Practical Experience with a Software Quality Assurance Program," *Comm. ACM,* 25, 1 (1982), 4.

Halloran, D., S. Manchester, J. Moriarty, R. Riley, J. Rohrman, and T. Skramstad, "Systems Development Quality Control," *MIS Quart.*, 2, 4 (1978), 1–13.

Hansen, J. V., "Audit Considerations in Distributed Processing Systems," *Comm. ACM*, 26, 8 (1983), 562.

Keidel, R. W., "Theme Appreciation as a Construct for Organizational Change," *Management Science*, 27, 11 (1981) 1261–1278.

Kriebel, C. H. and A. Raviv, "An Economics Approach to Modeling the Productivity of Computer Systems," *Management Science*, 26, 3 (March 1980), 297–311.

Kull, D. "Anatomy of a 4GL Disaster," *Computer Decisions*, (February 11, 1986) 58–65.

Leland, H. E., "Quacks, Lemons and Licensing: A Theory of Minimum Quality Standards," *J. Political Economy*, v. 87, 6 (1979), 1328–1346.

Lientz, B. P. and E. B. Swanson, *Software Maintenance Management: A Study of the Maintenance of Computer Application Software in 487 Data Processing Organizations*, Addison-Wesley, Reading, MA, 1980.

McFarlan, W. F., "Management Audit of the EDP Department," *Harvard Bus. Rev.*, (May–June 1973) 131–138.

Merten, A., and D. G. Severance, "Data Processing Control: A State-of-the-Art Survey of Attitudes and Concerns of DP Executives," *MIS Quart.*, 5, 2 (1981), 11–32.

Mendelson, H., "Pricing Computer Services: Queueing Effects," *Comm. ACM*, 28, 3 (March 1985) 312–321.

Milgrom, P. R. and J. Roberts, "Informational Asymmetries, Strategic Behavior and Industrial Organization," *American Economic Review, Papers and Proceedings*, 77 (1987), 184–193.

Milne, F. and R. Weber, "The Economics of Designing Generalized Software," *Comm. ACM*, 26, 8 (August 1983), 582.

Nolan, R. L., "Managing the Crises in Data Processing," *Harvard Bus. Rev.* (March-April 1979), 115–123.

Olson, M., *The Logic of Collective Action*, Harvard Univeristy Press, Cambridge, MA, 1965.

Rittenberg, L., and Charles R. Purdy, "The Internal Auditor's Role in MIS Developments," *MIS Quart.*, 2, 4 (1978), 47–57.

Schlender, B. R., "How to Break the Software Logjam," *Fortune*, (September 25, 1989), 100–112.

Simunac, D., "The Pricing of Audit Services: Theory and Evidence," *J. Accounting Research*, 18 (1980), 161–190.

Spence, A. M., "Job Market Signaling," *Quart. J. Economics*, (August 1973), 355–374.

Stigler, G. J., "The Economics of Information," *J. Political Economy*, 69 (1961), 213–225.

Swanson, E. B., *Information System Implementation*, Irwin, Homewood, IL (1988), 58–73.

Varian, H. R., *Microeconomic Analysis, 2nd Ed.*, Norton, New York, NY (1984), 80–81.

Westland, J. C. "A Net Benefits Approach to Measuring Retrieval Performance," *Information Processing and Management*, 25, 5 (October 1989), 579–581.

Westland, J. C., "Scaling Up Output Volumes Predicted by Information Systems Prototypes," *ACM Transactions on Database Systems*, September (1990a).

Westland, J. C., "Economics Constraints in Hypertext," *J. American Society for Information Science*, September (1990b).

Westland, J. C., "Topic Specific Monopolies in the Information Services Industry: Evidence from the DIALOG Group of Database," *The Information Society*, 6 (1990c), 127–138.