Total Quality Management in Information Systems Development: Key Constructs and Relationships

T. RAVICHANDRAN AND ARUN RAI

T. RAVICHANDRAN is an Assistant Professor in the Lally School of Management and Technology at Rensselaer Polytechnic Institute. He received his doctorate from the Southern Illinois University at Carbondale in 1996. He has a master's degree in industrial and systems engineering from the National Productivity Council of India and a bachelor's degree in production engineering from the University of Madras, India. His research interests include management of systems delivery, software quality and reusability, technology management, and organizational impacts of information technology. His research has been published or is forthcoming in *Communications of the ACM, European Journal of Information Systems, MIS Quarterly,* and the proceedings of several national and international conferences. He is a member of the Academy of Management, Association for Information Systems, Decision Sciences Institute and INFORMS.

ARUN RAI is an Associate Professor in the Electronic Commerce Institute at Georgia State University. He was a faculty member in the Department of Management at Southern Illinois University at Carbondale for seven years before joining Georgia State in the Fall of 1997. He received his Ph.D. from Kent State University in 1990. His present research interests include the diffusion, infusion, and impacts of information technology, emergent models of *e*-business, information technology design for information and knowledge management, and management of unstructured processes such as innovation, product development, decision making, and systems development. Dr. Rai has published articles in journals such as *Annals of Operations Research, Accounting, Management and Information Technologies, Communications of the ACM, Decision Sciences, Decision Support Systems, European Journal of Information Systems, Journal of Management Information Systems, Omega,* and several others. He is an associate editor of *MIS Quarterly* and *Information Resources Management Journal.* He is a member of the Academy of Management, Association for Computing Machinery, Decision Sciences Institute, INFORMS, Beta Gamma Sigma and Phi Kappa Phi.

ABSTRACT: The availability of high-quality software is critical for the effective use of information technology in organizations. Research in software quality has focused largely on the technical aspects of quality improvement, while limited attention has been paid to the organizational and sociobehavioral aspects of quality management. This study represents one effort at addressing this void in the information systems literature. The quality and systems development literatures are synthesized to develop eleven quality management constructs and two quality performance constructs. Scales for these constructs are empirically validated using data collected from a national survey of IS organizations. A LISREL framework is used to test the reliability and validity of the thirteen constructs. The results provide support for the reliabil-

Journal of Management Information Systems / Winter 1999–2000, Vol. 16, No. 3, pp. 119–155

0742–1222 / 2000 $9.50 + 0.00.

ity and validity of the constructs. A cluster analysis of the data was conducted to examine patterns of association between quality management practices and quality performance. The results suggest that higher levels of institutionalization of all quality management practices are associated with higher levels of quality performance. Our results also suggest that key factors that differentiated high- and low-quality performing IS units include senior management leadership, mechanisms to promote learning and the management infrastructure of the IS unit. Future research efforts directed at causally interrelating the quality management practices should lead to the development of a theory of quality management in systems development.

KEY WORDS AND PHRASES: information systems management, software quality, systems development, total quality management theory.

INFORMATION TECHNOLOGY (IT) PLAYS A CRITICAL ROLE IN ORGANIZATIONS today. IT capabilities are being used to reengineer traditional business processes and to redesign organizations for today's information-intensive environment. Systems development is a key process on which the success or failure of such IT-based organizational transformation efforts depends. Rockart and Hofman [81] observe that in many organizations systems development is "not only on the critical path to getting new products or services to market, it is the stumbling block on that path" (p. 21).

Recurrent problems such as poor system quality, long development lead time, user dissatisfaction, and high costs indicate that development of systems under acceptable conditions of quality and productivity remains a challenge. The escalating demand for new systems is expected to further compound quality problems in systems development. This concern is reflected in studies that indicate that systems development quality is a key issue facing IS executives in the 1990s [69, 89].

The attention to quality improvement is driven by increased organizational dependence on information systems and the magnitude of potential losses associated with poor systems quality. For example, software bugs in the baggage handling system delayed the opening of the Denver International Airport, which resulted in a $1.1 million per day increase in operating costs [38]. In addition to business losses, poor systems quality leads to project failures and cancelations. After spending $6 million, the Internal Revenue Service canceled its tax system modernization project partly because of quality problems. The Standish Group estimates that, in 1995 alone, more than $81 billion was wasted by U.S. companies in projects that ended up delivering poor-quality systems [91].

Managing systems development quality involves two fundamental but interdependent dimensions. First, a sophisticated technological infrastructure that enables the design and construction of high quality systems needs to be implemented. Second, an organizational system that enables analysts and programmers to engage in quality-oriented behavior needs to be established. Systems quality research has predominantly focused on the first dimension, which has led to increased understanding of the technical aspects of systems quality management. On the other hand, fundamental understanding of the organizational drivers of systems quality has not grown at the same pace. Given that the challenges facing IS development performance improvement are largely organizational and not technical in nature, it is imperative that research is directed at the organizational dimensions of quality management.

Total Quality Management (TQM), with its emphasis on the organizational and sociobehavioral aspects of quality improvement, can add to existing research on systems quality management. TQM is an integrated management philosophy that has been found to strongly influence organizational performance [34, 44]. Organizations are beginning to realize that TQM principles can be applied beyond manufacturing. In particular, the IS function presents many opportunities for TQM. Organizations such as Corning, Inc. [88] and Dun and Bradstreet Software [59] have found that TQM practices such as empowerment and benchmarking have a positive impact on systems development performance. Others have reported the positive effects of software process improvement on software quality, cost, and schedule adherence [61, 62].

While TQM practices have been selectively applied to IS development, they have not been elaborated upon and integrated with other systems development practices. Furthermore, a holistic analysis of the application of TQM concepts to systems quality management has not been undertaken. Systematic application of TQM theory to IS development is required to develop a better understanding of the organizational drivers of systems quality. Our objectives for this paper include synthesizing the IS, software engineering, and TQM literatures to identify and define key constructs of TQM in systems development; developing scales for TQM constructs in systems development and empirically testing their reliability and validity; examining patterns of association among the quality management constructs and quality performance; and identifying the relative importance of the identified quality management factors in making the transition from low-quality-performing to high-quality-performing IS development organizations.

This paper first identifies quality management themes previously discussed in the IS and software engineering literatures and examines their linkages with TQM concepts. We also identify the gaps and shortcomings in current research on systems quality management and examine how TQM concepts can enrich and extend systems quality research. Next, we synthesize the TQM literature to identify key TQM constructs, define these constructs in the context of systems development, and develop scales for each construct. We then discuss the empirical study and the iterative process adopted for construct validation and scale refinement and examine the patterns of association between the quality management practices and quality performance. Finally, we identify and discuss some directions for the development of a theory of quality management in systems development.

Literature Review

A SYSTEMATIC REVIEW OF THE IS AND SOFTWARE ENGINEERING LITERATURES indicates that selected TQM concepts have been applied to investigate the information systems quality phenomenon, sometimes without explicit reference to or linkage with the total quality management literature. Past research on the IS quality phenomenon has predominantly focused on techniques and tools for software quality assurance, the quality impacts of software process innovations and design methodologies, and development process management. We summarize significant research in each of these areas and examine the linkages with relevant TQM concepts. We use our synopsis of the literature to identify shortcomings and gaps in the management of systems development quality, thereby motivating our present work.

Based on an extensive review of the software quality assurance research, Rai, Song, and Troutt [76] conclude that researchers have emphasized software quality characteristics, software metrics, and quality control techniques and tools. TQM techniques, such as statistical quality control and quality function deployment, have been adapted and applied to software development [92, 108]. Some studies have empirically investigated the impact of these techniques on software quality outcomes [2, 16, 70, 72]. While these measurement and analytical techniques have been found to be useful in tracking and controlling specific quality problems, their impact on system quality depends on effectively linking individual product and process metrics to broader system quality objectives [100]. Limited research has been undertaken to develop measurement frameworks that link quality objectives to process and product metrics. Furthermore, quality control techniques are unlikely to be effective unless they are an integral part of an organizational system for quality improvement.

A large body of system quality research has conceptualized systems development as a technical process emphasizing precision and technical accuracy in design and construction. Research on software process innovations, such as CASE and reusability, suggests that these tools and techniques have a positive effect on code quality and programmer productivity. [13, 82, 104, 3, 11, 60]. However, their effect on overall system quality has been marginal because a large proportion of quality problems originates during requirement definition and system design [105]. Furthermore, studies also conclude that organizations face major hurdles in the implementation of software process innovations and that these hurdles are organizational, not technological, in nature.

Process improvement is a core TQM concept. The Software Engineering Institute (SEI) has developed specific models to evaluate, diagnose, and evolve the capabilities of the development process. SEI's Capability Maturity Model (CMM) defines an evolutionary path from *ad hoc*, chaotic processes to mature, disciplined processes. Process maturation, as assessed by the predictability of development outcomes in terms of budget, schedules, and quality, is enhanced when feedback is meaningfully generated and used to recalibrate and fine-tune process design. Anecdotal evidence suggests that organizations implementing CMM-based software process improvement have realized gains in development cycle time and programmer productivity [28, 41, 45]. Reports also suggest that organizations face difficulties in adhering to the sequence of change implementation recommended by CMM [17, 74, 83].

Process improvement is one aspect of TQM that needs to be integrated with other core TQM principles such as customer focus and viewing the organization with an integrated systems perspective. These essential aspects of TQM, which are currently missing in the CMM, can play a major role in enabling or constraining process management efforts [73]. The omission of key organizational factors from the CMM and the lack of theory informing the conceptualization of the CMM stages raise questions about the rationale for the suggested sequencing to develop process capabilities.

In summary, while previous research on IS development has examined some important TQM concepts, key gaps exist in this area of research. First, an integrated analysis of the application of TQM concepts to information systems development has not been undertaken. Consequently, no coherent theory of systems quality management has emerged. Second, systems quality research has focused on the technical and engineering aspects of quality control, while paying limited attention to the organizational dimensions of quality management. Third, a systemic perspective of quality management is lacking in current IS research. Our objective is to move toward a theory of information systems quality management. Identification of constructs and scale development is an important first step in meaningful theory development in systems quality management. Accordingly, we synthesize the TQM, systems development, and software engineering literatures to identify and define key quality management constructs and develop scales for them. In the next section, we review the TQM literature to identify the critical factors of quality management and in the subsequent section we adapt these factors to the systems development context.

Critical Factors of Quality Management

THE QUALITY MANAGEMENT LITERATURE IS LARGELY BASED on the prescriptions of recognized quality proponents, including Deming, Crosby, Juran, Feigenbaum, and Ishikawa. Recent studies in the areas of operations management and strategic management have synthesized these prescriptions and identified critical factors of quality management [1, 33, 84]. Based on a detailed literature review, Saraph, Bensen, and Schroeder [84] identified top management leadership for quality, the role of the quality department, training, product/service design, supplier quality management, process management, quality data and reporting, and employee relations as eight important quality management constructs. They used the data collected from 162 managers to validate the proposed measurement scales for these constructs. Flynn, Schroeder, and Sakakibara [33] synthesized the practitioner literature and government reports on quality and identified additional constructs such as quality improvement rewards and work-force management. They used the data collected from multiple respondents in forty-two manufacturing plants to validate their constructs. More recently, Ahire at al. [1] extended the work of Saraph et al. [84] and Flynn et al. [33] and identified additional constructs, such as customer focus and SPC usage. They validated their constructs using the data collected from 371 manufacturing firms in the United States. Table 1 summarizes the constructs examined in these three studies.

Key Constructs of TQM in Systems Development

WE ADAPTED THE CRITICAL FACTORS OF TQM TO THE SYSTEMS DEVELOPMENT context. Nunnally [71] suggested that, when borrowing constructs from reference disciplines, it is imperative that researchers closely examine the meaning of each construct within the context in which it was first defined, and then evaluate if the meaning holds true in a different context. We followed a systematic process of mapping the TQM factors to the IS development context and identified eleven quality management and two quality performance constructs. The identified constructs are also summarized in Table 1.

Constructs are latent variables that cannot be directly observed. Hence, it is necessary to measure the manifestation of a construct using items that constitute a scale. The items for each construct were identified by directly adapting the items from the validated quality management scales [1, 33, 84] to the systems development context. Where feasible, items from existing scales in the IS literature were used in the development of the operational measures of the quality management and quality performance constructs. The operational measures of the thirteen scales are shown in the appendix. We discuss each of these constructs and their measurement scales.

IS Management Commitment to Quality

Deming [27] asserts that without senior management's commitment to quality improvement and visible signaling of their commitment, an organization will not be able to change its practices that lead to poor quality. In fact, top management commitment to quality is a common factor to all quality management frameworks [22, 27, 56, 85, 87]. Empirical studies support the thesis that top management commitment to quality promotes quality-oriented practices and behaviors in the organization [6, 34, 84]. These findings are consistent with transformational leadership theories [12, 97], which suggest that senior management can encourage change by formulating and communicating a vision for the future and reinforcing values that support this vision. Senior management may demonstrate confidence and moral conviction in their values [46], espouse an appealing vision that generates enthusiasm for certain value-laden ideological goals [20, 97] and serve as role models for the value system [99]. Senior management manifests their commitment by their personal involvement in activities such as quality planning and performance review, ownership of responsibility for quality performance, and providing support for quality initiatives [10, 27].

Saraph et al. [84] propose a scale for "divisional management leadership and quality policy" to assess management support for quality. This scale combines senior management responsibilities and policy issues into one scale, which has been critiqued in the quality literature [1, 33]. We used only the items pertaining to senior management responsibilities from this scale and adapted them to the IS context to develop the scale for the IS Management Support for Quality construct.

Quality Policy and Goals

Policies provide broad guidelines for decision making and a framework for prioritizing goals. Explicit policies can stress the strategic importance of quality and focus

Table 1. Summary of Quality Management and Quality Performance Constructs

Saraph et al. [84]	Flynn et al. [33]	Ahire et al. [1]	Our study
Quality management constructs			
Top management leadership and quality policy	Top management support	Top management commitment	IS management commitment to quality
	Quality policy not explicitly considered	Not considered	Quality policy and goals
Nature of reward schemes included under employee relations	Considered under top management support	Considered under employee involvement but dropped from the validated scale	Quality orientation of reward schemes
Training	Included under work-force management	Employee training	Commitment for skill development
Product/service design	Product design	Design quality management	Formalization of analysis and design; formalization of reusability in systems development
Quality data and reporting	Quality information	Internal quality information usage; benchmarking	Fact-based management
Process management	Process management	SPC usage	Process control
Employee relations	Work-force management	Employee empowerment; employee involvement	Empowerment of programmer/analyst
Customer involvement not explicitly considered	Customer involvement	Customer focus	User participation
Supplier quality management	Supplier involvement	Supplier performance	Vendor/consultant participation
Quality performance constructs			
Not explicitly considered	Product quality in terms of scrap rate	Product quality	Product quality
Process quality not explicitly considered as a performance measure	Process quality not explicitly considered as a performance measure	Process quality not explicitly considered as a performance measure	Process efficiency

organizational members' attention on quality goals. Quality-oriented policies provide a mechanism to implement the philosophy that quality receives a higher priority over cost or schedule and that in the long run superior quality leads to improvements in cost and delivery performance [56]. Established quality policies have to be communicated to organizational members through specific quality goals. Techniques such as policy deployment have been used by Japanese organizations to translate quality objectives into a hierarchy of goals [49]. Furthermore, steps must be taken to ensure that organizational members comprehend quality goals and the means to achieve these goals.

Saraph et al. [84] enumerated key practices that are useful in identifying whether an organization has explicit quality policies or not. These include specificity of quality goals, comprehensiveness of the goal-setting process, importance attached to quality in relation to other goals, and the extent to which quality goals are reviewed and their attainment is emphasized. However, they conceptualized these practices as a subdimension of leadership. This approach fails to make a conceptual distinction between providing a vision and the mechanisms for translating the vision into action—an important distinction made in the leadership literature [97]. We adapted the policy-related items from Saraph et al. [84] to the systems development context to develop the scale for the Quality Policy and Goals construct.

Quality Orientation of Reward Schemes

Performance metrics and reward schemes used by an organization reflect issues considered important by management. Traditionally, rewards for analysts and programmers have been based on task efficiency, with measures such as lines of code or function points produced used to assess their productivity. These measures overlook process effectiveness. Consequently, rewards based on such measures can hamper adoption of innovative practices. Thus, changes to reward structures may be necessary to promote quality-oriented behavior among systems development teams [59, 88]. In a case study at Corning Inc., Shrednick et al. [88] found that incentives for spending within budget, customer satisfaction, process improvement, and cost reduction resulted in significant improvements in the quality of services provided by IS teams. Kane [59] found that Dun and Bradstreet Software incorporated performance contingent rewards to drive improvement of their software development process. Drawing on the case studies reported in the IS literature and adapting the items from Flynn et al. [33] to an IS context, we developed a scale for the Quality Orientation of Reward Schemes construct.

Commitment to Skill Development

Core principles of quality management, such as customer focus and continuous improvement, depart significantly from traditional management practices [39, 77]. Quality-oriented training programs can facilitate organizational members' understanding of change initiatives and can influence their attitudes toward change. Some recommendations have been made about the specific orientation of quality-oriented training programs. Deming [27] argued that employees should develop a meaningful understanding of analytical tools and techniques and their applications in the context of quality problems. Training in group dynamics, team building, and problem-solving skills needs to be incorporated into a holistic training program [33]. In addition, the IS literature stresses that IS personnel should have good business skills in order to be effective in systems development tasks.

Commitment to skill development pertains to the extent to which an IS unit is interested in developing the capabilities of IS personnel. This is reflected in the extent to which training is provided to IS personnel on an ongoing basis and in the extent to which the IS unit is ready to commit resources for skill development. Our survey of the IS literature revealed that, while much attention has been paid to skill development of IS personnel, scales to assess the IS unit's commitment to skill development have not been developed. Thus, we adapted the items from Flynn et al. [33] to the systems development context to propose a scale for the Commitment to Skill Development construct.

Formalization of Analysis and Design

The manufacturing literature underscores the importance of product design in managing product quality [37, 96]. A large proportion of product failures is due to design weakness [19], which can be reduced with explicit attention to design quality [37, 96]. Poor understanding of customer requirements is a major source of design problems [87]. Formal techniques such as concurrent engineering, design for manufacturability, and quality function deployment have been developed to better understand customer needs and engineer product quality.

The systems development literature also notes that a significant proportion of quality problems arise in the initial phases of development, such as requirements analysis and design [105]. Several formal analysis and design techniques have been proposed to facilitate effective requirement determination and the translation of identified user requirements into systems design. Techniques such as Joint Application Design and prototyping increase communication between users and developers and compress the analysis-testing cycle. By structuring tasks and their interrelationships, formal design methods and techniques reduce the complexity of the design process. For example, the rules and procedures specified by design techniques, such as data modeling, process modeling, and interaction modeling [48], reduce task complexity by encoding general problem-solving knowledge and heuristics, such as "divide and conquer," and by providing a framework for the organization of tasks. The importance of using formal analysis and design techniques to achieve project objectives is now well documented in the systems development literature [103], although measurement scales to assess the use of these techniques have not been developed. Drawing from the prescriptions for effective systems development, we developed a scale for the Formalization of Analysis and Design construct.

Formalization of Reusability in Systems Development

Systems development has long been treated as a craft, and a job-shop approach to system construction has been followed. This approach has prevented organizations from exploiting a strategy commonly used in other engineering processes: the design and development of interchangeable components [24]. Increasingly, IS units are realizing that a factory approach focusing on developing reusable components can increase development quality, decrease cost, and reduce lead time. Field studies indicate that reusability has enabled Japanese and U.S. firms to achieve significant improvements in productivity and quality [3, 23, 54, 55, 93]. These studies indicate that formal policies can promote construction of reusable components and mandate component usage across projects [24, 60]. These studies also identify specific actions that can be taken by organizations to formalize reusability. These actions include recognition of reusability as a corporate objective, institutionalization of corporation-wide efforts to plan for reuse, and systematic monitoring of reuse within and across development projects [60].

Cusumano [24] developed an instrument to assess reuse in software development firms. We synthesized the anecdotal evidence presented in the IS literature with the scale items from Cusumano [24] to develop a scale for the Formalization of Reusability construct.

Fact-Based Management

The objective of fact-based management is to uncover sources of quality problems and process wastes, and to understand cause-and-effect relationships between process parameters and process outcomes. Information acquisition through measurement is central to fact-based management. Deming [27] stresses the importance of extracting the information inherent in process variations through the use of statistical techniques. Taguchi [95] extended the role of information acquisition in quality management by suggesting that experiments must be designed to measure and determine causes of quality problems precisely. The Baldrige Award criteria explicitly include a category for information management that emphasizes systematic collection and analysis of information obtained from internal and external sources. Close scrutiny of the award criteria suggests that almost all the factors in the Baldrige framework include collection, synthesis, and dissemination of information at various levels in the organization. Reimann [78], a Baldrige Award inspector, points out that "the most common factor among companies scoring high in the evaluation process is that they had instituted systematic measurement processes" (p. 63).

Researchers in systems development literature have also emphasized the importance of measurement in process improvement. For example, the Capability Maturity Model focuses on the relationship between the characteristics of measurement systems and the capability of the development process. The model describes five levels of maturity with the sophistication of metrics increasing with the level of maturity [47]. A carefully conceptualized measurement system serves as a framework for defining shared goals and relating process level goals with the overall goals of the IS unit. As part of such a measurement system, metrics should be selected with great care to ensure consistency between goals and measures. Periodic recalibration of metrics may be required to ensure their validity over time [100].

Data collected through measurement should be analyzed to learn how quality objectives can be met and exceeded. At the operational level, data analysis involves identifying deviations from expected norms and revealing causes for these deviations. At higher levels, data analysis involves detecting and making sense out of patterns. By adapting the scale items from Saraph et al. [84] to the systems development context, we developed a scale for the Fact-Based Management construct that reflects the collection and use of quality-related information.

Process Control

Performance standards can be used to define targets for improvement [49]. By definition, continuous improvement requires that performance standards be dynamic and constantly revised. The *plan-do-check-act* (PDCA) cycle systematizes the use of standards for performance control and improvement. The cycle starts with planning where targets for improvement are set. Existing operational problems, process variations, or opportunities for performance enhancement form the basis for these plans. The do and check phases involve plan implementation and performance evaluation. If satisfactory results are achieved, the new methods or processes are standardized and routinized. If problems are encountered, the plans are systematically evaluated based on new insights learned in the do and check phases. This dynamic process is repeated until satisfactory results are achieved. A major spinoff benefit of this cycle is the process knowledge that is generated. This process knowledge is sharable across projects, creating a ripple effect in performance improvement. Consistent with the continuous improvement literature, we developed a scale that reflects the notion of Process Control using dynamic performance standards.

Empowerment of Programmer/Analyst

Participation of programmers and system analysts in core development tasks is inherent in their defined organizational roles. However, participation in administrative and project management tasks depends on the control structure of development teams. Two typical structures that have been proposed are chief programmer teams [65] and egoless programming [101]. In the former, the decision-making authority is centralized with the chief programmer, while in the latter control is diffused throughout the team. Yourdon [106] argues against centralized control and points out that one individual would not have the capability to handle the communication and decision-making complexity involved in any systems development project. While others have pointed out that decentralized control could lead to schedule and cost overruns in projects [15, 65], the positive impacts of analyst and programmer participation in certain administrative tasks are acknowledged. We conceptualize programmer/analyst empowerment as the extent to which they take part in project management activities and decisions.

Giving employees more autonomy and control over their work and decisions related to their work could lead to initiation and persistence of behaviors oriented toward task accomplishment [20]. Such an empowerment process allows leaders to set higher performance goals, which are accepted by employees. Empowerment may also be useful in motivating subordinates to persist despite difficult organizational/environmental obstacles. Furthermore, deep understanding of work processes embedded among organizational members is an important, often untapped resource for project management. Such knowledge is tacit, and it is present in the skills and mental models of people in the organization. Many Japanese organizations successfully use empowerment to tap into the tacit knowledge of their employees and put it to productive use [70].

The information systems literature also discusses the positive effects of programmer/analyst participation in project management. Cost and schedule adherence was found to be high in projects where design team members had control over their outputs [43]. The same study also found that restricting managerial control to setting behavior expectations from the team led to better team performance. Shrednick et al. [88] discuss how empowerment of IS teams at Corning, Inc. dramatically improved service levels and user satisfaction and reduced costs. Based on the empirical evidence in the IS literature, we developed a scale for the Programmer/Analyst Empowerment construct.

User Participation

Researchers and IS professionals agree that systems development efforts are unlikely to be successful without active user participation in the development process [51, 57]. User participation can enhance the conformance of system features to their needs [64]. Factors such as improved user and developer understanding of the system, improved assessment of user needs, and user feedback on system features account for the positive relationship between user participation and system quality [51, 68].

User participation is usually passive where systems analysts interact with users to elicit system requirements. Such participation has very limited impact on systems development quality. On the other hand, users can be part of the IS team and play an active role in the design process and share responsibilities in systems development. The nature of participation should be determined by the complementarity between the knowledge and skills that users and developers can bring to development tasks. Since active user participation increases the scope for role conflicts between analysts and users [79, 80], the nature of participation should be matched with task requirements. Almost all of the past studies emphasize the importance of user participation for purposes of requirement analysis and design. Since 72 percent of errors in software are attributed to poor specifications and design [105], active user participation during the early phases of the development life cycle is critical. Since users are directly involved in providing inputs to the systems and using system outputs, user participation in defining the content and format of system inputs and outputs should be beneficial. User participation in system testing could also be very valuable. Intimate knowledge of the problem domain is required for formulating meaningful test plans. Not surprisingly, software vendors have been increasingly beta-testing their products with actual users in their day-to-day work contexts. We used the items from existing scales in the IS literature [51, 79] to operationalize the User Participation construct.

Vendor Participation

Participation of suppliers in the core design and production processes is receiving increasing attention in the quality literature. Such participation promotes the development of mutual understanding of design and production constraints (faced by both the organization and its vendors) in enhancing quality and meeting customer needs. Strategies such as making vendors an integral part of organizational processes, investing in enhancement of vendors' quality performance, and building long-term partnerships are suggested as mechanisms to promote the reliability and quality of parts supplied. Some empirical evidence also exists to support the position that effective management of vendor relationships is critical to improve quality performance [33]. In fact, one of the key evaluation criteria in the Baldrige Award framework pertains to the extent to which policies and systems have been instituted to foster active involvement of vendors in organizational processes.

Vendor and consultant participation in systems development is not new among IS units. However, some organizations are moving away from traditional project-based contracting to longer-term relationships with external agents, as project-based contracts provide limited incentives for external agents to invest in the organization's long-term quality improvement initiatives [63]. Vendor participation can also be used to lower knowledge barriers associated with the deployment of specific systems development techniques, technologies, and methodologies [8]. However, it is necessary that vendors/consultants and IS personnel work closely for effective knowledge transfer to occur. Such close interaction can be achieved only when vendors and/or consultants form an integral part of the systems delivery process. We developed a scale for the Vendor Participation construct that reflects the extent to which vendors are an integral part of the development process and the extent to which long-term partnerships have been developed with vendors.

Quality Performance

Product Quality

Key system quality dimensions including portability, reliability, efficiency, human engineering, and maintainability have been identified in the software engineering literature. A variety of metrics to assess these dimensions of system quality have also been developed and validated. While this stream of research continues to evolve, its emphasis has been on the engineering characteristics of the software and limited attention has been paid to assessing and enhancing users' subjective evaluations of the software.

A key management objective when dealing with information products, including software, is to understand the value placed by users on these products [86]. In contrast to the technical focus of software quality assurance research, customer satisfaction is an important objective of TQM initiatives. Customers have specific requirements, and products/services that effectively meet these needs are perceived to be of higher quality [27, 56]. A similar perspective is evident in the IS management literature as significant attention has been paid to understanding user requirements and satisfying them. Significant research attention has been directed at identifying the dimensions of user satisfaction and developing reliable and valid instruments for the measurement of this construct [9, 36, 52].

User satisfaction is considered a valid measure of systems quality as it reflects users' subjective evaluation of the features and functionality of the information system [35]. In fact, many scholars argue that, since a system's prime requirement is to serve users, their perceptions are a very important measure of system quality [58]. Moreover, Shapiro and Varian [86] note that the quality of information-based products, such as software, should be a function of the product's utility to the users and that user satisfaction is the only system quality measure that reasonably reflects users' utility function regarding a system.

From a user's perspective, the critical attributes of an information system include: (1) the usefulness of the functionality of the system, (2) the extent to which the system provides relevant and timely information, and (3) the extent to which the system possesses superior engineering-oriented performance characteristics [26, 42]. We developed a scale for the Product Quality construct that reflects these critical system attributes from the users' perspective.

Process Efficiency

Process measures of quality are equally important from a customer's perspective as it bears a relation to the cost of goods and services and their efficient delivery. Product quality cannot be thought of apart from product cost [32]. From a customer's perspective, availability, price, and convenience are factors that complement product quality in the sense that they focus on the process of product/service delivery and reflect the efficiencies of these processes. Thus, process efficiency is an important dimension of quality performance.

In the IS literature, aspects such as "being on time, within budget and meeting user needs" [66], and elimination of waste due to rework and errors [30] have been used to define process efficiency. The underlying theme of these definitions focuses on two key aspects: (1) effective resource utilization and (2) elimination of non-value-adding activities in the process. These two aspects are reflected in our scale for the Process Efficiency construct.

Empirical Study

Survey Instrument Development

A CROSS-SECTIONAL NATIONAL SURVEY WAS CONDUCTED TO COLLECT THE DATA for the study. Following recommendations for developing survey instruments [71], we used a seven-point Likert scale to ensure statistical variability among survey responses for all but one construct, "formalization of reusability in systems development." This construct was measured using a five-point scale, which was adapted from Cusumano [24]. The survey instrument was pilot-tested with two IS executives, two software quality consultants, and four IS researchers working in systems development. Suggestions made by the respondents were incorporated and a final version of the instrument was developed.

Sampling and Data Collection

The population of interest is IS units that develop application systems in-house. We limited our sampling frame to IS organizations in *Fortune* 1,000 companies and large government agencies. We followed a systematic approach in constructing the mailing list for the survey. First, the *Fortune* 1,000 organizations were identified through a search of the *Compustat* corporate database. Organizations such as holding companies, conglomerates, and trusts were dropped from the mailing list. This yielded a set of 700 organizations. Next, the mailing addresses for these organizations were obtained from the *Directory of Top Computer Executives* [29]. Organizations not listed in the directory were dropped, resulting in a set of 605 *Fortune* 1,000 companies. Finally, 105 federal and state government agencies were randomly chosen from the same directory to construct the total sample for the study.

Senior IS executives were chosen as the respondents as they are likely to be most informed about quality initiatives in IS units. The names of senior IS executives in the sampled organizations were identified from the *Directory of Top Computer Executives* [29]. Where multiple names were found, the most senior person was chosen as the respondent. A total of 710 questionnaires were mailed. Four mailings, each spaced apart by three weeks, were undertaken. One hundred and twenty-three usable responses were received, resulting in a response rate of 17.32 percent (Table 2).

The response rate is modest, but close to the minimum recommended level of 20 percent for organizational surveys [40, 107] and similar to those obtained in many IS surveys [73]. Pinsonneault and Kraemer [75] found that more than two-thirds of IS surveys at the organizational level had a sample size of smaller than 150 and an overwhelming majority of IS surveys had a low response rate. The difficulties in obtaining high response rates for IS surveys could be partly attributed to the sheer number of surveys targeted at IS managers. Given the challenges associated with surveying IS managers, the response rate to our survey appears reasonable and can be interpreted as indicative of the interest in the survey theme among IS managers.

Table 2. Profile of Respondents by Industry

Industry	Effective no. of questionnaires mailed	No. of responses received	Response Rate (%)
Manufacturing	338	64	18.93
Insurance	34	6	17.65
Utilities	34	6	17.65
Transportation	29	6	13.79
Retail	32	5	15.63
Banks	61	8	13.11
Financial services	25	5	20.00
Div. services	52	5	9.62
Government	105	18	17.14
Total	710	123	17.32

Nevertheless, it is recommended that all efforts be made to maximize response rates and reduce the chances of sampling error [107].

We took several steps to mitigate the chances of sampling error. First, we provided incentives (such as a summary of the survey results and a pack of coffee) to respondents and conducted multiple mailings to improve our response rate (to the current level of 17.32 percent). Second, we polled nonrespondents to assess the reasons for nonresponse and to check if factors specific to our study accounted for the modest response rate. Finally, we systematically checked for nonresponse bias by comparing respondents with nonrespondents.

A telephone poll of sixty randomly chosen nonrespondents was conducted. A standard protocol was developed to structure the telephone conversations so as to ensure that the questions posed to the participants were similar. The questions focused on the reasons for nonresponse, the relevance of our survey to the organization, and whether the organization had adopted TQM in its IS units. The major reasons for nonresponse indicated were: (1) the large number of surveys received by them (53.3 percent), (2) company policy not to respond to surveys (13.1 percent), (3) length of the questionnaire (16.6 percent), (4) lack of interest in the survey theme (8.3 percent), and (5) lack of time due to other commitments (such as organizational restructuring) (8.3 percent). These results suggest that the significant reasons for nonresponse are not specific to this study and represent a more general trend. However, it is likely that the length of our survey instrument could have deterred a small proportion (16 percent) of the surveyed population from participating. Furthermore, 38 percent of these sixty nonrespondents polled indicated that they had not adopted TQM practices in their IS units. While we polled only sixty nonrespondents, it appears that nonadopters of TQM may have been more likely not to respond to our questionnaire, raising some cautionary implications for the external validity of our findings.

Proportionate classification of respondents and nonrespondents was compared on key organizational characteristics such as industry (SIC codes), organization size, and annual revenue. The chi-square analysis provided evidence of the absence of response bias. Table 2 indicates that the response rate did not vary much across industry segments, providing further evidence of the absence of response bias. In addition to comparing respondents and nonrespondents, comparison of early and late respondents is recommended. The respondents were split into three equal groups based on their response date. One-way ANOVA was used to test for differences between the first (early respondents) and the third (late respondents) groups on a variety of demographic variables, such as industry, organization size, ISD size, and time since adoption of quality management practices. No significant differences were detected between the groups, suggesting that the respondents can be pooled with no loss in generalizability.

The firms that responded represent a broad cross-section in terms of industry, organization size, and ISD size: 52.03 percent of the respondents were manufacturing firms, 33.33 percent were service organizations, and 14.64 percent were government agencies; 21.7 percent of the firms had 500 or fewer employees, 32.5 percent had between 500 and 5,000 employees, 40 percent had more than 5,000 employees (median 3,900 employees); and 25 percent of the firms had 50 or fewer employees in their information systems units, 15 percent had between 50 and 100 employees, 20 percent had between 100 and 200 employees, and 40 percent had more than 200 employees (median 137 employees). The respondents were senior IS executives (director of MIS, 62.4 percent; CIO, 21.3 percent; vice president, MIS, 12.4 percent), and 82 percent of them were within two levels from the CEO in the organizational hierarchy.

Confirmatory Factor Analysis

We used confirmatory factor analysis, as implemented within the LISREL framework [53], to validate the eleven quality management and two quality performance constructs. This allowed us to specify a measurement model consisting of a construct defined according to the weighted linear combination of its indicators, and to assess the fit of the specified measurement model to the data. Such a specification subscribes to a causal-indicator model where the observed indicators reflect the unobserved theoretical construct. Typically, a causal-indicator model is specified and analyzed for each theoretical construct individually [1, 98]. We followed these guidelines for all constructs with four or more indicators. Constructs with fewer indicators were pooled and analyzed in order to provide adequate degrees of freedom for estimation of model parameters. In our study, three constructs (IS commitment to quality, empowerment of programmer/analysts, user participation) have three items, and one construct (vendor participation) has two items. Items for these four constructs were pooled and analyzed, resulting in a model that was overidentified with 38 degrees of freedom.[1]

Following the guidelines for scale validation [4, 14, 18], a series of analyses was done to assess unidimensionality, reliability, convergent validity, discriminant validity of the quality management and quality performance constructs, and the criterion-related validity of the quality management constructs.

Table 3. Assessment of Unidimensionality, Reliability and Convergent Validity

Construct	No. of items	Unidimensionality Goodness of fit index [GFI]	Reliability Cronbach's α	Reliability Werts Linn Jorsekog ρ_c	Convergent validity Bentler Bonnet Δ
1. IS management commitment to quality	3†	0.94	0.79	0.80	0.92
2. Quality policy and goals	5	0.96	0.84	0.85	0.95
3. Commitment to skill development	4	0.99	0.70	0.71	0.98
4. Quality orientation of reward schemes	4	0.97	0.68	0.79	0.91
5. Formalization of reusability in systems development	4	0.99	0.85	0.86	0.99
6. Formalization of analysis/design	4	0.95	0.77	0.78	0.90
7. Fact-based management	8	0.93	0.87	0.87	0.92
8. Process control	5	0.92	0.95	0.95	0.96
9. User participation	3†	0.94	0.78	0.86	0.92
10. Programmer/ analyst empowerment	3†	0.94	0.65	0.67	0.92
11. Vendor/ consultant participation	2†	0.94	0.71	0.77	0.92
12. Product quality	4	0.90	0.82	0.83	0.87
13. Process efficiency	4	0.96	0.78	0.79	0.95

† A combined model was run for these four constructs.

Unidimensionality

Unidimensionality is a necessary prerequisite for reliability and validity analyses [71]. A construct is unidimensional if its constituent items represent one underlying trait. In confirmatory factor analysis, specifying a measurement model that defines the relationship between each construct and its constituent items tests unidimensionality. A good fit of the measurement model to the data indicates that, as hypothesized, all items load significantly on one underlying latent variable. The fit of the measurement model is indicated by the goodness of fit index. The GFI indices for all thirteen constructs are higher than the recommended level of 0.90 (Table 3). These results suggest that all thirteen scales are unidimensional.

Reliability

Reliability can be defined as the degree to which measures are free from error and, therefore, yield consistent results. Operationally, reliability is defined as the internal consistency of a scale, which assesses the degree to which the items are homogeneous. Cronbach's alpha is a widely used measure of internal consistency [21, 71]. A scale is considered reliable if the alpha coefficient is greater than 0.70. The composite reliability measure proposed by Werts, Linn, and Joreskög [53], which is an alternate conceptualization of reliability, represents the proportion of measure variance attributable to the underlying trait. The Werts, Linn, and Joreskög ρ_c represents the ratio of trait variance to the sum of trait and error variance. Scales with ρ_c greater than 50 percent are considered to be reliable.

Both tests were used to assess the reliability of the thirteen scales. Table 3 indicates that the ρ_c values are well above the threshold of 0.5 for all scales. The Cronbach's alpha values were also found to be greater than 0.70 for all but two scales (quality orientation of reward schemes and programmer/analyst empowerment). We did not refine these two scales further for three reasons. First, the alpha values for both scales were close to the cutoff value of 0.70 (0.68 and 0.65) and greater than the minimum recommended (0.60) for newly developed scales [71]. Second, the ρ_c values for both scales were greater than 0.50. Third, dropping items would yield scales that may not adequately sample the domain of the constructs. We do recommend that the operational measures of these two constructs be carefully examined before future studies use them.

Convergent Validity

Convergent validity is the extent to which varying approaches to construct measurement yield the same results. A commonly used method to assess convergent validity is to view each item in a scale as a different approach to measure the construct [1]. Convergent validity is then checked using the Bentler-Bonnet coefficient (Δ). The Bentler-Bonnet coefficient represents the ratio of the chi-square value of the specified measurement model to that of a null model, which has no hypothesized item loadings on a construct. Scales with Δ values of 0.90 or above demonstrate strong convergent validity. The Bentler-Bonnet coefficients for all the thirteen scales are given in Table 3. Except for product quality, the Δ values are greater than 0.90 for all scales. Although the Δ value for product quality (0.87) is lesser than 0.90, it is close to the threshold value.

Discriminant Validity

Discriminant validity refers to the degree to which measures of different constructs are unique from each other. This is achieved when measures of each dimension converge on their corresponding true scores and do not converge on true scores of other constructs. The following procedure is followed for assessing discriminant validity: Confirmatory factor analysis is run on pairs of scales, allowing for correlation between the constructs. Next, the procedure is repeated with the correlation between the two constructs constrained to be equal to 1. A significant difference between the constrained model chi-square and that of the unconstrained model indicates that the two constructs are distinct [1, 98].

Discriminant validity checks were run for all pairs of the eleven quality management constructs and the two quality performance constructs. This resulted in a total of fifty-six tests.[2] The chi-square difference test was found significant ($p < 0.001$) for all fifty-six tests, indicating discriminant validity among the eleven quality management scales and the two quality performance scales.

Criterion-Related Validity

Criterion-related validity refers to the extent to which constructs predict theoretically related outcome variables. Specifically, criterion-related validity pertains to the extent to which the quality management constructs are related to measures of quality performance. This is an important component of construct assessment because it moves the logic of assessment from the "statistical domain of intercorrelations among the multiple indicators underlying a trait to the substantive domain focusing on relationships that are best interpreted in the light of theory" [98, p. 954].

We examined the relationship between each quality management construct and quality performance. In addition to the quality performance measures developed here (product quality and process efficiency), we also used a measure of software process maturity in assessing the criterion-related validity of the quality management constructs. Software process maturity reflects the extent to which systems development process parameters are optimized to enhance process effectiveness and the extent to which the process is in control. The Software Engineering Institute's Capability Maturity Model [47] depicts five levels of software process maturity ranging from "initial" (an *ad hoc* and chaotic systems development process) to "optimized" (a systems development process under control and continuously improved through measurement and feedback). Theoretical support for including process maturity as a dimension of quality performance can be found in the well-accepted notion that a capable process is a necessary prerequisite for delivering quality products and services [27, 50, 87]. We used an unidimensional response matrix enumerating the five maturity levels along with their descriptions to measure process maturity (see the appendix for the response matrix). The descriptions of the maturity levels were borrowed directly from those given in the Capability Maturity Model [47].

Table 4 summarizes the eleven tests carried out to relate each quality management construct with product quality, process efficiency, and process maturity. All thirty-three relationships were in the expected direction; twenty-eight of these relationships were significant, providing evidence of the criterion-related validity of the respective quality management constructs. Formalization of reusability in systems development and vendor/consultant participation were not significantly related to product quality and process efficiency, and user participation was not significantly related to process maturity. However, the associations were in the expected directions providing some evidence of the criterion-related validity of these three constructs. Furthermore, each of these three constructs was significantly related to one or two of the three criterion-related measures considered.

Table 4. Assessment of Criterion-Related Validity

	Quality performance					
	Product quality		Process maturity		Process maturity	
Construct	γ	*t*-value	γ	*t*-value	γ	*t*-value
1. IS management commitment to quality	0.201	10.911**	0.240	20.402*	0.951	20.232*
2. Quality policy and goals	0.255	20.432*	0.170	10.689**	0.974	30.220*
3. Commitment to skill development	0.218	20.037*	0.232	20.278*	0.0.956	20.318*
4. Quality orientation of reward schemes	0.466	40.097*	0.394	30.650*	0.0.991	50.443*
5. Formalization of reusability in systems development	0.140	10.354	0.071	00.721	0.985	40.411*
6. Formalization of analysis/design	0.279	20.532*	0.204	10.938**	0.0.984	30.922*
7. Fact-based management	0.463	40.475*	0.312	30.193*	0.980	30.697*
8. Process control	0.434	40.355*	0.217	20.266*	0.975	30.428*
9. User participation	0.346	30.395*	0.127	10.290	0.695	00.717
10. Programmer/analyst empowerment	0.580	50.127*	0.414	30.880*	0.992	50.545*
11. Vendor/consultant participation	0.107	10.076	0.089	00.886	0.961	20.309*

* $p < 0.01$; ** $p < 0.05$.
γ represents the path coefficient in the structural model.

Relationships Among the Constructs

Systemic Pattern of Association Among the TQM Constructs

WHAT ARE THE PATTERNS OF ASSOCIATION OF THE QUALITY MANAGEMENT practices among themselves and with quality performance? Are IS organizations achieving high levels of quality performance by implementing a small subset of these eleven practices? Or do these eleven practices represent an overall system of quality management that needs to be put into place? We used cluster analysis to answer these questions. This involved deriving distinct and meaningful clusters of the eleven quality management practices and examining the levels of the three quality performance measures associated with each of these clusters.

The IS organizations represented in our database were cluster-analyzed over the eleven quality management practices using a nonhierarchical procedure. The mean

Table 5. Clustering IS Units on TQM Practices

Quality management/quality performance	Cluster 1 ($n = 47$)	Cluster 2 ($n = 47$)	Cluster 3 ($n = 25$)	1 – 2 p	1 – 3 p	2– 3 p
Quality management practices						
IS management commitment to quality	5.92	6.20	3.91	ns	0.000	0.000
Quality policy and goals	4.60	5.21	2.98	0.000	0.0023	0.000
Quality orientation of rewards	3.12	4.41	2.75	0.000	ns	0.000
IS commitment to skill development	4.50	5.61	3.80	0.000	0.008	0.000
Formalization of design methods	4.10	5.63	4.08	0.000	ns	0.000
Formalization of reusability	2.78	4.06	2.52	0.000	ns	0.000
Fact-based management	3.73	4.70	2.66	0.000	0.000	0.000
Process control	4.22	5.26	2.70	0.000	0.000	0.000
Programmer/analyst empowerment	5.00	5.75	4.79	0.000	ns	0.000
User participation	5.45	6.29	5.44	0.000	ns	0.001
Vendor participation	4.87	5.50	3.82	0.006	0.004	0.000
Quality performance						
Process maturity	3.87	4.59	3.64	0.003	ns	0.001
Process efficiency	3.66	4.37	3.25	0.007	ns	0.000
Product quality	5.00	5.54	4.72	0.012	ns	0.007

values for each of the quality performance measures were computed for each of the clusters. Statistical significance of the differences between the mean values of each of the quality management constructs and quality performance measures across the three clusters were examined using t-tests.

As shown in Table 5, the analysis yielded three clusters. Cluster 2 comprises mainly manufacturing firms with relatively high TQM experience and quality performance. Clusters 1 and 3 comprise firms with relatively low TQM experience and success. Of these, cluster 3 consists primarily of service organizations, while cluster 1 consists of manufacturing organizations.

An examination of the mean values of each of the clusters reveals some interesting patterns. Cluster 2 has the highest mean values for all quality management factors. This cluster also has the highest mean values for all three quality performance measures. Cluster 1 has the second highest set of values for all quality management practices. This cluster also has the second highest set of mean values for the three quality performance measures. Finally, cluster 3 has the lowest set of values for all quality management practices and for the three quality performance measures.

For each of the eleven TQM practices and for the three quality performance variables, we assessed the significance level of the differences in mean values between the extracted clusters. In all, forty-two tests were conducted to evaluate if, in fact, the mean values for the variables significantly differed between clusters. The risk of inflating the probability of a Type 1 error due to multiple testing, such as we are doing here, requires the use of more stringent alpha levels [94]. Accordingly, we established an overall probability of a Type 1 error for our tests at an alpha level of 0.05, and, consequently, examined each of the forty-two tests at a level of significance of 0.05/42, which is an alpha level of 0.0011. Even at this very low level of significance, all eleven TQM factors are observed to differ between the low-quality cluster (cluster 3) and the high-quality cluster (cluster 2). Furthermore, the mean values for process maturity and process efficiency also differ at this level of significance, while differences in mean values for product quality approach this level of significance with a p value of 0.007. Tabachnik and Fidell [94] note that in multiple testing situations a higher alpha can be established for variables considered especially important, as is the case with product quality. We can therefore conclude with a very high degree of confidence that low- and high-quality performers differ with respect to the identified TQM practices. It is clear from our results that organizations with high levels of quality performance were found to have higher levels for all quality management constructs in comparison with organizations with lower levels of quality performance. Our results are consistent with conclusions reached by scholars in operations management and strategic management that TQM is an integrated strategy for organizational performance improvement that is supported by a coherent set of mutually reinforcing practices [25, 90].

Relative Importance of TQM Practices

What is the *relative* importance of these TQM practices in enabling a transition to high-quality-performing IS organizations in terms of systems development? To answer this question, we rank-ordered the eleven TQM factors based on the effect size, as measured by the t-statistic, associated with mean differences between the low-quality-performing (cluster 3) and high-quality-performing (cluster 2) groups. Our results are summarized in Table 6. Several interesting insights emerge from the rank-ordering of the TQM factors.

Leadership, as assessed by IS management commitment to quality, ranks on top of the list as a differentiating factor between the low and high quality performance IS organizations. IS units that have high-quality performance appear to have a senior IS management that is committed to quality improvement. On the other hand, IS units with low-quality performance may be continuing to frame quality as a technical issue, and relegating quality management to lower levels within the IS organization, such as programmers/analysts or project managers. This observation is consistent with empirical findings that, in successful IS organizations, senior IS managers envi-

Table 6. Relative Importance of the Quality Management Practices in Discriminating Between Low-Quality-Performing and High-Quality-Performing IS Units

Quality management and quality performance constructs	Difference between cluster 2 and cluster 3 (t-values)
IS management commitment to quality	11.399
Fact-based management	10.109
Process control	9.524
Quality policy and goals	9.192
IS commitment to skill development	8.693
Quality orientation of rewards	7.620
Formalization of reusability	7.171
Formalization of design methods	5.892
Vendor participation	5.807
Programmer/analyst empowerment	4.457
User participation	4.148
Quality performance	
Product quality	3.740
Process maturity	3.489
Process efficiency	3.114

sion and create superior organizational systems to manage core IS processes such as systems development [31, 81].

The next two variables in terms of their rank order represent themes of learning and continuous improvement. Improved quality performance indicates a state where a fundamental understanding of the causes of poor quality exists and that these causes are being eliminated much in the vein of continuous improvement. Development of such an understanding requires both first-order and second-order learning [7]. Process control through performance standards promotes first-order learning, as it encourages sustained attempts to maintain or exceed desired performance goals. These performance goals could include cost, quality, delivery, and productivity standards to be maintained by development teams. The resultant task knowledge facilitates achievement of quality goals within the constraints of a defined development process. On the other hand, second-order learning is required to rethink the development process. The systematic collection and use of quality data as well as the sharing of knowledge among analysts, users, and vendors/consultants facilitate identification of fundamental causes of quality problems and promote a deeper understanding of the process drivers of quality. A deeper understanding of the process drivers of quality is useful in sustaining quality improvements through fundamental changes to the development process. Thus, it can be expected that IS units exhibiting higher-quality performance will have effectively institutionalized practices such as fact-based management and process control.

The next three variables, as assessed by their ranks, represent what we call management infrastructure practices. Quality policy and goals, quality orientation of rewards, and IS commitment to skill development collectively establish the management context within which development processes are defined and improved. A clear statement of goals, aligning reward schemes with cooperative group work, and providing requisite skill sets to employees to operate in a TQM environment define a sophisticated management infrastructure. Such an infrastructure should ideally promote the conduct of day-to-day development processes and build motivation and commitment among people closest to the everyday activities of systems development. In fact, empirical studies have found that improvements to the development process through software process innovations are more likely to be successful if an enabling management context is first created [3, 60].

Formalization of reusability and design methods rank after the management infrastructure variables. Participation by vendors, programmer/analyst empowerment, and user participation rank last in terms of variables that differentiate low- and high-quality performers. This is not to say that these five variables are not important, but that in comparison with the other variables examined here, they seem to be practiced relatively more uniformly across IS organizations today. Among these variables, reusability is diffused to a lesser extent than the other practices, as is evident from its low mean values among all three clusters.

The relative rank order of the TQM practices reveals an interesting pattern. The IS literature emphasizes the formalization of design methods and how these interventions can enhance systems development performance. Similarly, since the early 1980s, the IS literature has advocated the importance of user participation and examined the benefits of empowered structures, such as when programmer/analysts are operating in "egoless" teams. It appears that these practices have diffused significantly through the population of IS organizations we surveyed. Our analyses also revealed that these practices were positively associated with quality performance indicating that these practices may be necessary for the success of systems development efforts. However, they clearly are not a basis for defining a TQM-based IS organization. Our rankings of TQM practices suggest that leadership for quality is critical, as are the mechanisms for learning and continuous improvement. Furthermore, management infrastructure practices set the context for technical interventions at the process level as well as day-to-day development activities.

Discussion

IN THIS STUDY, THE QUALITY AND SYSTEMS DEVELOPMENT LITERATURES were synthesized to identify and define eleven quality management constructs and two quality performance constructs. Scales were developed for all thirteen constructs. Confirmatory factor analysis was used to validate the scales. The results can be interpreted as providing support for the reliability and validity of all thirteen scales. It must be noted that developing valid scales transcends a single study. Researchers are encouraged to question the conceptual and operational definition of the constructs proposed here and to examine the validity and reliability of these constructs using different data. It must also be noted that constructs such as user participation are not new to IS researchers. Valid scales for user participation have been developed, and the relationship between this construct and systems development performance has been examined in a variety of research contexts. Similarly, themes underlying formalization of analysis and design and empowerment of programmer/analysts have been examined in the IS literature. However, these constructs have not been collectively conceptualized in terms of their possible implications for system development quality. By identifying and defining a comprehensive set of interrelated TQM constructs, this study permits future researchers to use common definitions and assumptions to study the relationships between quality management and quality performance. Furthermore, this study extends IS research on systems quality management by synthesizing concepts well established in the IS literature with new ones borrowed from TQM theory.

Implications for Research: Directions for Theory Building

The constructs developed here provide the building blocks for a theory of quality management in systems development. One has to be cautious, however, before proceeding to hypothesize relationships between the TQM constructs. Premature formulation and testing of hypotheses could lead to erroneous conclusions if any moderating, mediating, or second-order relationships were present among the quality management constructs. In the context of developing a TQM theory, Anderson et al. [5] evaluated higher-order constructs and then proceeded to examine their interrelationships. Similarly, future research can examine second-order constructs that can be established from the first-order quality management constructs and then interrelate these higher-order quality management constructs into an integrative framework for systems quality improvement.

We propose that leadership for IS quality, management infrastructure sophistication, development process management efficacy, and stakeholder participation in the development process are second-order factors that are defined by the first-order factors identified earlier. An inspection of the correlations in Table 7 provides some preliminary support for the existence of these second-order factors. IS management commitment to quality can be considered an important reflector of quality-oriented leadership. The high correlations of quality policy with quality-oriented rewards (0.38) and commitment to skill development (0.52) suggest that these three constructs are formative indicators of a quality-oriented management infrastructure. Organizations that have adopted these practices can be characterized as having a sophisticated management infrastructure and hence would be better prepared to redesign, formalize, and manage the systems development process. Conversely, organizations that have not adopted these practices have a less sophisticated management infrastructure and hence lack the infrastructure context to implement process-level changes to achieve quality outcomes. Similarly, the correlations between formalization of reusability, formalization of design methods (0.40), fact-based management (0.44), and process control (0.37) support the notion that these first-order constructs are formative indicators of process management efficacy. These practices are integral to pro-

Table 7. Correlation Among Quality Management and Quality Performance Constructs

	QM1	QM2	QM3	QM4	QM5	QM6	QM7	QM8	QM9	QM10	QM11	QP1	QP2
IS management commitment to quality (QM1)	1.00												
Quality policy and goals (QM2)	0.62	1.00											
Quality orientation of reward schemes (QM3)	0.29	0.33	1.00										
Commitment to skill development (QM4)	0.39	0.48	0.49	1.00									
Formalization of analysis/design (QM5)	0.27	0.30	0.41	0.46	1.00								
Formalization of reusability in systems development (QM6)	0.26	0.35	0.29	0.27	0.40	1.00							
Fact-based management (QM7)	0.53	0.64	0.60	0.46	0.46	0.44	1.00						
Process control (QM8)	0.39	0.44	0.43	0.39	0.48	0.37	0.62	1.00					
Programmer/analyst empowerment (QM9)	0.18	0.15	0.36	0.37	0.39	0.22	0.35	0.40	1.00				
User participation (QM10)	0.23	0.13	0.29	0.32	0.42	0.27	0.42	0.33	0.46	1.00			
Vendor participation (QM11)	0.35	0.27	0.27	0.29	0.25	0.26	0.34	0.38	0.29	0.27	1.00		
Product quality (QP1)	0.14	0.22	0.30	0.15	0.19	0.18	0.42	0.43	0.38	0.35	0.07	1.00	
Process efficiency (QP2)	0.16	0.20	0.38	0.27	0.20	0.15	0.34	0.25	0.29	0.15	0.02	0.48	1.00
Process maturity (QP3)	0.18	0.27	0.36	0.22	0.31	0.42	0.33	0.32	0.40	0.13	0.20	0.35	0.36

cess management and, as they are implemented, their effects can be traced in the extent to which the systems development process emphasizes efficiency, control, and adaptation, which are important prerequisites for the development of quality systems [47]. Finally, the correlations between employee empowerment, user participation (0.46) and vendor participation (0.29) suggest that these three constructs are formative indicators of stakeholder participation.

How do the higher-order constructs theoretically affect quality performance? We suggest one possible theoretical framework for investigating the relationships among the higher-order quality management constructs. Senior IS management leadership acts as a driver for quality management; its commitment must be translated into strategies for developing an organizational system for quality. This involves creating a quality-oriented management infrastructure, implementing process management practices, and motivating key stakeholders to engage in quality-oriented behavior. Elements of management infrastructure facilitate implementation of process-level changes, which include formalization of the systems development process and managing the process through systematic information collection and use. Furthermore, a quality-oriented infrastructure should foster active participation of key stakeholders, such as programmers/analysts, users, and vendors in systems development. Stakeholder participation is required to identify and eliminate sources of quality problems and thereby improve systems quality. Together with stakeholder participation, process management practices result in a mature systems development process and continuous improvement of product quality and process efficiency.

We do not claim, nor is it the intent of this paper to prescribe a specific theory of systems development quality management. However, the above framework represents one of the many possible theoretical perspectives that need to be developed into a theoretical model and tested in future research. This framework should be compared with other potentially competing theoretical perspectives. As a continuation of our research in this area, we are in the process of drawing upon appropriate theoretical perspectives, including the one briefly sketched out here, to develop and test alternative theoretical models for the management of systems development quality.

Implications for Practice

Our findings have several implications for IS managers. We found a systemic pattern of association among the quality management constructs, suggesting that all identified factors are important in improving quality performance. Thus, a coherent, integrated strategy encompassing adoption of all identified dimensions of quality management is required, as opposed to the implementation of one tool or management practice. IS units embarking on a TQM program should carefully examine their infrastructures to assess whether the explicit and implicit policies in the organization, reward systems, and skill development programs enable or constrain the design of effective processes or the extraction of the best from people involved in day-to-day development. Attempting to implement process-level changes by formalizing design methods and reusability is important, but inadequate in the absence of an overall organizational context, in establishing a high-quality IS development organization. Desirable stakeholder behaviors, such as user participation, and their outcome-related consequences are better understood today. These behaviors are likely to be promoted by the organizational context, but are unlikely to be very effective if other organizational variables, such as leadership and management infrastructure, are not aligned to support high-quality performance. Thus, IS managers are well advised to frame quality improvement as an organizational change program and to direct attention at managing the transition to a TQM environment.

Limitations

We used a key informant method for data collection. Both quality management and quality performance data were collected from senior IS managers. It is possible that the self-reported quality performance measures could be biased. However, these measures represent the perceptions of IS executives who most likely are responsible for championing TQM and sanctioning resources for the TQM initiatives. Their perceptions of product quality and process efficiency will therefore be an important factor that influences TQM adoption and implementation.

Our measures of quality performance are based on responses of IS managers and represent perceptions of IS managers in the aggregate about product quality, process maturity, and process efficiency. We compared the self-reported process maturity levels with the results of diagnostic surveys conducted by SEI. The distribution of the self-reported maturity levels across organizations matched very closely with those published by SEI. This provides some confidence that the self-reported quality performance measures are reasonable estimates of quality performance. Nevertheless, we suggest that future researchers consider two alternatives in assessing quality performance. First, objective quality measures could be used in addition to the perceived measures used here. Second, quality performance can be assessed by surveying end users instead of IS managers.

We used a two-item scale to measure vendor participation. This scale may not have adequately covered the domain of this construct. We recommend that future research should further develop the vendor participation construct with a focus on lowering technical knowledge barriers associated with new development tools and methods, and the integration of technical knowledge embodied in these tools with the systems of organizations adopting these tools.

Conclusions

LACK OF ESTABLISHED THEORIES ON SYSTEMS QUALITY MANAGEMENT motivated us to undertake this study. In the light of increasing pressures on IS managers to improve systems quality and the growing importance of quality management within the IS function, this study is both timely and significant. Because it is a scale development endeavor, it should be considered as setting the stage for further work in this domain. While many IS researchers have called for adopting a sociotechnical perspective in systems development research, systems quality management has largely been studied from a technical perspective. This study is one of the early attempts to identify and define sociobehavioral and organizational factors that should be considered by IS researchers. It provides evidence that systems quality performance cannot be improved by piecemeal adoption of quality management practices and stresses the need for a coherent strategy of implementing an organizational system for quality improvement.

Many IS units might have effectively implemented selected practices, such as user participation and formalization of design methods, and may still be facing development quality problems. The growing interest in software process improvement suggests that process management practices are increasingly being adopted by IS units. Our analysis revealed that while these practices are important for improving systems quality performance, they are unlikely to be very effective in the absence of other identified TQM practices. In fact, we believe that the difficulties encountered by organizations in implementing CMM-based process improvements can be partly attributed to the limited attention paid in the CMM model to the organizational drivers of quality including the management infrastructure of IS units and IS management leadership. Thus, IS units are well advised to adopt an integrated strategy encompassing adoption of all identified dimensions of TQM, as opposed to the implementation of one tool or management practice.

Notes

Acknowledgment: The authors would like to acknowledge the partial funding provided for this project by the Pontikes Center for the Management of Information at the Southern Illinois University at Carbondale.

1. Degrees of freedom pertains to the number of bits of information available for estimating the sampling distribution of the data after all model parameters have been estimated. Measurement models with three indicator variables are just identified with one degree of freedom and will yield a perfect fit. Models with fewer indicators are underidentified and will always yield incorrect loadings. One approach to overcome both these problems is to pool the indicators for underidentified and just identified constructs and specify a combined measurement model.

2. Number of paired tests = $N_1C_m + N_2C_m = \{N_1!/[(N_1-m)! * m!]\} + \{N_2!/[(N_2-m)! * m!]\}$
$= 11!/[9! * 2!] + 2!/[0! * 2!] = 56$ tests.

References

1. Ahire, S. L.; Golhar, D.Y.; and Waller, A.M. Development and validation of TQM implementation constructs. *Decision Science, 27*, 1 (1996), 23–56.

2. Ahituv, N., and Zelek, M. Instant quality control of large batch processing jobs. *MIS Quarterly, 11*, 3(1987), 313–323.

3. Apte, U.; Sankar, S.C.; Takur, M.; and Turner.E.J. Reusability-based strategy for development of information systems: implementation experience of a bank. *MIS Quarterly, 14*, 4 (1990), 421–433.

4. Anderson, J.C., and Gerbing, D. W. Structural equation modeling in practice: a review and recommended two-step approach. *Psychological Bulletin, 103*, 3 (1988), 411–423.

5. Anderson, J.C.; Rungtusanthanam, M.; and Schroeder, R. A theory of quality management underlying the Deming management method. *Academy of Management Review, 19*, 3 (1994), 473–509.

6. Anderson, J.C.; Rungtusanthanam, M.; Schroeder, R.; and Devaraj, S. A path analytic model of a theory of quality management underlying the Deming management method: preliminary empirical findings. *Decision Sciences, 26*, 5 (1995), 637–658.

7. Argyris, C., and Schon, D. A. *Organizational Learning: A Theory of Action Perspective*. Reading, MA: Addision Wesley, 1978.

8. Attewell, P. Technology diffusion an organizational learning: the case of business computing. *Organization Science, 3*, 1 (1992), 1–19.

9. Bailey, J.E., and Pearson, S. W. Development of a tool for measuring and analyzing computer user satisfaction. *Management Science, 29*, 5 (1983), 519–529.

10. Baldrige Award—Malcolm Baldrige National Quality Award. Washington, DC: National Institute of Standards and Technology, 1992.

11. Banker, R.D., and Kauffman, R.J. Reuse and productivity in integrated computer-aided software engineering—an empirical study. *MIS Quarterly, 15*, 3 (1991), 375–401.

12. Bass, B. M. Leadership and Performance Beyond Expectations. New York: Free Press, 1985.

13. Bendure C.O. A case study on CASE: its evolution and use at HHMI. *Journal of Information Systems Management, 8*, 4 (1991), 50–56.

14. Bollen, K.A. *Structural Equations with Latent Variables*. New York: John Wiley, 1989.

15. Brooks, F.P. *The Mythical Man-month*. MA: Addison-Wesley, 1975.

16. Camuffo, M.; Maiocchi, M.; and Morselli, M. Automatic software test generation. *Information and Software Technology, 32*, 2 (1990), 337–346.

17. Card, D. Understanding process improvement. *IEEE Software, 8* (July 1991), 102–103.

18. Chin, W.W. Issues and opinion on structural equation modeling. *MIS Quarterly, 22*, 1 (1998), vii–xvi.

19. Cole, R. E. The Japanese lesson in quality. *Technology Review, 83*, 7 (1981), 29–40.

20. Conger, J.A., and Kanungo, R.N. The empowerment process: integrating theory and practice. *Academy of Management Review, 13*, 3 (1987), 471–482.

21. Cronbach, L.J. Coefficient alpha and the internal structure of tests. *Psychometrika, 16* (September 1951), 297–334.

22. Crosby, P. *Quality Is Free*. New York: Mentor Publishing, 1979.

23. Cusumano, M.A., and Kemerer.C.F. A quantitative analysis of US and Japanese practice and performance in software development. *Management Science, 36*, 11 (1990), 1384–1406.

24. Cusumano, M.A. *Japan's Software Factories*. Oxford: Oxford University Press, 1991.

25. Dean, J.W., and Bowen, D.E. Management theory and total quality: improving research and practice through theory development. *Academy of Management Review, 19*, 3 (1994), 387–418.

26. DeLone, W.H., and McLean, E.R. Information systems success: the quest for the dependent variable. *Information Systems Research, 3*, 1 (1992), 60–95.

27. Deming, E. *Out of the Crisis*. Cambridge, MA: MIT Center for Advanced Engineering, 1986.

28. Diaz, M., and Sligo, J. How software process improvement helped Motorola. *IEEE Software, 14*, 5 (1997), 75–81.

29. *Directory of Top Computer Executives*. Phoenix, AZ: Applied Research Inc., 1994.

30. Eilon, S. Measuring quality of information systems. *Omega, 21*, 2 (1993), 135–138.

31. Feeney, D.F.; Edwards, B.R.; and Simpson, K.M. Understanding the CEO/CIO relationship. *MIS Quarterly, 16*, 4 (1992), 434–447.

32. Feigenbaum, A.V. *Total Quality Control*, 4th ed. New York: McGraw-Hill, 1991.

33. Flynn, B.; Schroeder, R.G.; and Sakakibara, S. A framework for quality management research and an associated measurement instrument. *Journal of Operations Management, 11*, 4 (1994), 339–366.

34. Flynn, B.; Schroeder, R.G.; and Sakakibara, S. The impact of quality management practices on performance and competitive advantage. *Decision Sciences, 26*, 5 (1995), 659–692.

35. Franz, C.R., and Robey, D. Organizational context, user involvement and the usefulness of information systems. *Decision Sciences, 17*, 3 (1986), 329–355.

36. Galletta, F.G., and Lederer, A.L. Some cautions on the measurement of user information satisfaction. *Decision Sciences, 20*, 3 (1989), 419–438.

37. Garvin, D.A. Competing on the eight dimensions of quality. *Harvard Business Review, 65*, 6 (1987), 101–109.

38. Gibbs, W.W. Software crisis. *Scientific American, 271*, 3 (1994), 86–95.

39. Grant, R.M.; Shani, R.; and Krishnan, R. TQM's challenge to management theory and practice. *Sloan Management Review, 35*, 2 (Winter 1994), 25–35.

40. Grover, V. A tutorial on survey research: from constructs to theory. http://theweb.badm.sc.edu/grover/survey/MIS-SUVY.html (1997).

41. Haley, T.J. Raytheon's experience in software process improvement. *IEEE Software, 13*, 2 (1996), 33–41.

42. Hamilton, S., and Chervany, N., L. Evaluating information systems effectiveness-comparing evaluation approaches. *MIS Quarterly, 5*, 3 (1981), 55–69.

43. Henderson. J.C., and Lee. S. Managing IS design teams: a control theories perspective. *Management Science, 38*, 6 (1992), 757–777.

44. Hendricks, K.B., and Singhal, V.R. Does implementing an effective TQM program actually improve operating performance? empirical evidence from firms that have won quality awards. *Management Science, 43*, 9 (1997), 1258–1274.

45. Hollenbach, C.; Young, R.; Pflugrad, A.; and Smith, D. Combining quality and software process improvement. *Communications of the ACM, 40*, 6 (1997), 41–45.

46. House, R.J. A theory of charismatic leadership. In J.G. Hunt, and L.L Larson (eds.), *Leadership: The Cutting Edge*. Carbondale: Southern Illinois University Press, 1976, pp. 189–207.

47. Humphrey, W.S. Characterizing the software process: a maturity framework. *IEEE Software, 5*, 2 (March 1988), 62–75.

48. IEF. *A Guide to Information Engineering Using the IEF*, 2d ed. Dallas: Texas Instruments Incorporated, 1988.

49. Imai, M. *Kaizen*. New York: Random House, 1986.

50. Ishikawa, K. *Guide to Quality Control*. Tokyo: Asian Productivity Organization, 1976.

51. Ives, B., and Olson, M.H. User involvement in MIS success: a review of research. *Management Science, 30*, 5 (1984), 586–603.

52. Ives, B.; Olson, M.H.; and Baroudi, J.J. The measurement of user information satisfaction. *Communications of the ACM, 26*, 10 (1983), 785–793.

53. Joreskog, K.G., and Sorbom, D. *LISREL 7: A Guide to the Program and Applications*. Chicago: SPSS, Inc, 1988.

54. Jones, T.C. *Programming Productivity*. New York: McGraw-Hill, 1986.

55. Joyce, E.J. Reusable software: passage to productivity? *Datamation, 34*, 18 (1988), 97–102.

56. Juran, J.M. *Quality Control Handbook*. New York: McGraw-Hill, 1986.

57. Kaiser, K.M., and Srinivasan, A. The relationship of user attitudes toward design criteria and information system success. In *Proceedings of the National American Institute of the Decision Sciences*, 1980.

58. Kan, S.H.; Basili, V.R.; and Shapiro, L.N. Software quality: an overview from the perspective of total quality management. *IBM Systems Journal, 33*, 1 (1994), 4–19.

59. Kane, E.J. Implementing TQM at Dun and Bradstreet Software. *National Productivity Review, 11*, 3 (Summer 1992), 405–416.

60. Karimi, J. An asset-based systems development approach to software reusability. *MIS Quarterly, 14*, 2 (1990), 178–198.

61. Krishnan, M. S. Cost and quality considerations in software product management. Ph.D. dissertation, GSIA, Carnegie Mellon University, Pittsburgh, 1996.

62. Lawlis, P.K.; Flowe, R.M.; and Thordahl, J.B. A correlational study of the CMM and software development performance. *CrossTalk* (September 1995), 21–25.

63. Loh, L., and Venkatraman, N. Determinants of information technology outsourcing: a cross-sectional analysis. *Journal of Management Information Systems, 9*, 1 (1992), 7–24.

64. Lucas, H.C. System quality and the use of information systems. *Management Informatics, 3*, 4 (1978), 207–212.

65. Mantei, M. The effect of programming team structures on programming tasks. *Communications of the ACM, 24*, 3 (1981), 106–113.

66. Martin, J. *Information Engineering*. Cranforth, Lancashire, UK: Savant Research Studies, 1989.

67. Mumford, E. Consensus systems design: an evaluation of this approach. In N. Szyperski, and E. Grochla (eds.), *Design and Implementation of Computer Based Information Systems*. Gromingen: Sijthoff and Noordhoff, 1979.

68. Munson, J., and Khoshgoftaar, T. The detection of fault prone programs. *IEEE Transactions on Software Engineering, 18,* 5 (1992), 423–433.

69. Niederman, F.; Brancheau, J.C.; and Weatherbe, J. *Information systems management issues for the 1990s. MIS Quarterly, 15,* 4 (1991), 475–500.

70. Nonaka, I. The knowledge creating company. *Harvard Business Review, 69,* 6 (1991), 96–103.

71. Nunnally, J.C. *Psychometric Theory.* Englewood Cliffs, NJ: McGraw-Hill, 1988.

72. Okumoto, K.A. Statistical method for software quality control. *IEEE Transactions on Software Engineering, 11,* 12 (1985), 1424–1430.

73. Paulk, M.C.; Weber, C.V.; Curtis, B.; and Chrissis, M.B. *The Capability Maturity Model: Guidelines for Improving the Software Process.* Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 1995.

74. Pfleeger, S.L. Realities and rewards of software process improvement. *IEEE Software, 13,* 6 (1996), 99–101.

75. Pinsonneault, A., and Kraemer, K.L. Survey research in management information systems: an assessment. *Journal of Management Information Systems, 10,* 2 (1993), 75–105.

76. Rai, A., Song, H., and Troutt, M. Software quality assurance: an analytical survey and research prioritization. *Journal of Systems and Software, 40,* 1 (1998), 67–84.

77. Reger, R.K.; Gustafson, L.T.; DeMarie, R.; and Mullane, J. V. Reframing the organization: why implementing total quality is easier said than done. *Academy of Management Review, 19,* 3 (1994), 655–684.

78. Reimann, C.W. America unites behind the Baldrige Quality crusade. *Electronic Business* (October 15, 1990), 63.

79. Robey, D., and Franz, C. R. Group process and conflict in systems development. *Management Science, 35,* 10 (1989), 1172–1189.

80. Robey, D.; Franz, C.R.; Smith, L.A., and Vijayasarathy, L.R. Perception of conflict and success in information system development projects. *Journal of Management Information Systems, 10,* 1 (1993), 123–139.

81. Rockart, J.F., and Hofman, J.D. Systems delivery: evolving new strategies. *Sloan Management Review, 33,* 4 (Summer 1992), 57–64.

82. Rummens, N., and Sucher, R. CASE: competitive edge. *Systems International (UK), 17,* 3 (1989), 31–34.

83. Saiedian, H., and Kuzara, R. SEI capability maturity model's impact on contractors. *Computer, 28* (January 1995), 16–26.

84. Saraph, J.V.; Bensen, P.G.; and Schroeder, R.G. An instrument for measuring the critical factors of quality management. *Decision Sciences, 20,* 4 (1989), 810–829.

85. Schoenberger, R.J. *Total Quality Management: Nine Hidden Lessons in Simplicity.* Stamford, CT: Productivity Press, 1984.

86. Shapiro, C., and Varian, H.R. *Information Rules: A Strategic Guide to the Network Economy.* Boston: Harvard Business School Publishing, 1998.

87. Shingo, S. *Zero Quality Control: Source Inspection and Poka-Yoke Systems.* Stamford, CT: Productivity Press, 1986.

88. Shrednick, H.R.; Shut, R.J.; and Weiss, M. Empowerment: key to IS world-class quality. *MIS Quarterly, 16,* 4 (1992), 491- 505.

89. SIM quality assessment and planning tools for IS. Report of the Working Group on Quality, Society for Information Management, October 1992.

90. Sitkin, S.B.; Sutcliffe, K.M.; and Schroeder, R.G. Distinguishing control from learning in total quality management: a contingency perspective. *Academy of Management Review, 19,* 3 (1994), 537–564

91. The Standish Group Chaos. http://www.standishgroup.com/chaos.html (September 24, 1997).

92. Stylianou, A.; Kumar, R.; and Khouja, M.A. A TQM-based systems development process. *DATABASE for Advances in Information Systems, 28,* 3 (1997), 59–71.

93. Swanson, K.; McComb, D.; Smith, J.; and McCubbery, D. The application software factory: applying total quality techniques to system development. *MIS Quarterly, 15,* 4 (1991), 566–579.

94. Tabachnick, B.C., and Fidell, L.S. *Using Multivariate Statistics.* New York: HarperCollins, 1989.

95. Taguchi, G. *Introduction to Quality Engineering*. Tokyo: Asian Productivity Organization, 1986.

96. Taguchi, G., and Clausing, D. Robust quality. *Harvard Business Review, 68*, 1 (1990), 65–75.

97. Tichy, N.M., and Devanna, M.A. *The Transformational Leader*. New York: Wiley, 1986.

98. Venkatraman, N. Strategic orientation of business enterprises: the construct, dimensionality and measurement. *Management Science, 35*, 8 (1989), 942–962.

99. Waldman, D.A. The contribution of total quality management to a theory of work performance. *Academy of Management Review, 19*, 3 (1994), 510–536.

100. Walrad, C., and Moss, E. Measurement: the key to application development quality. *IBM Systems Journal, 32*, 3 (1993), 445–460.

101. Weinberg, G. *The Psychology of Computer Programming*. New York: Van Nostrand Reinhold, 1971.

102. Werts, C.E.; Linn, R.L.; and Joreskog, K.G. Interclass reliability estimates: testing structural assumptions. *Educational and Psychological Measurement, 34*, 1 (1974), 25–33.

103. Whitten, J.L.; Bentley, L.D.; and Barlow, V.M. *Systems Analysis and Design Methods*. Homewood, IL: Irwin Publishing, 1994.

104. Williamson, M. Getting a handle on CASE. *CIO, 3*, 7 (1990), 42–51.

105. Yates, W.D., and Shaller, D.A. Reliability engineering as applied to software. In *Proceedings of Reliability and Maintainability Symposium* (1990), 425–429.

106. Yourdon, E. *Managing the Structured Technique*. Englewood Cliffs, NJ: Prentice-Hall, 1976.

107. Yu, J., and Cooper, H. A qualitative review of research design effects on response rates to questionnaires. *Journal of Marketing Research, 20*, 1 (1983), 36–44.

108. Zultner, R. E. TQM for technical teams. *Communications of the ACM, 36*, 10 (1993), 79–91.

Appendix: Outline of the Questionnaire Used for the Study

All items except those for scale 6 solicit responses on a seven-point Likert scale with 1 = strongly disagree, 2 = disagree, 3 = disagree slightly, 4 = neutral, 5 = agree somewhat, 6 = agree, and 7 = strongly agree. Items for scale 6 (formalization of reusability in systems development) solicit responses on a five-point scale with 1 = none, 2 = low, 3 = moderate, 4 = high and 5 = very high. During data analysis this scale was normalized to a seven-point scale to maintain a uniform scale width for all constructs.

The following items pertain to quality management practices in your information systems department (ISD). For each item, please circle the choice that best indicates current practices in your ISD.

1. IS Management Commitment to Quality

   IS chief executive assumes responsibility for quality performance.

   IS chief executive is evaluated for quality performance.

   IS chief executive supports quality improvement processes.

2. Quality Policy and Goals

   IS management has clear quality objectives.

   Quality goals within IS are very specific.

   There is a comprehensive IS quality plan.

   Quality goals and policy are understood within the department.

   Significant importance is attached to quality in relation to cost and schedule objectives.

3. Quality Orientation of Reward Schemes

Development cycle time, cost and productivity are used as the basis for rewards for IS personnel.

User satisfaction is an important factor in determining rewards for IS personnel.

Quality measures like error rate and scrap rate are used as the basis for rewards for IS personnel.

Incentives are used to promote reusability.

4. Commitment to Skill Development

Regular training in quality management tools and techniques is given to IS personnel.

Team building and group dynamics training are given to IS personnel.

Business skills training is given to IS personnel.

Resources are made available for training IS personnel.

5. Formalization of Analysis and Design

Formal techniques such as JAD and prototyping are regularly used for requirement elicitation.

Idea generation techniques such as brain storming are used in system design.

Formal techniques such as quality function deployment are used to translate user requirements into design.

Standard representation schemes such as ER diagrams and DFD are used for design specifications.

6. Formalization of Reusability in Systems Development

Extent to which formal policies to promote development of reusable design/ code are implemented.

Extent to which formal policies that mandate use of reusable components are implemented.

Extent to which reuse of code/design components is monitored.

Extent to which formal policies on parameterization of design/code are implemented.

7. Fact-Based Management

Quality data are collected and reported at frequent intervals.

Vendors/consultants are pressed to furnish quality data.

Performance levels are benchmarked with those of other firms.

Quality problems are analyzed to identify problem causes.

Quality data are systematically used in managing systems development.

Cost of quality is analyzed.

Metrics are recaliberated to reflect changes in the development process.

Best practices are systematically institutionalized.

8. Process Control

Performance standards have been established for design.

Performance standards have been established for programming.

Performance standards have been established for testing.

Performance standards are used to monitor and control output.

Performance standards are revised annually/regularly.

9. User Participation

Users actively participate in determining system requirements.

Users actively participate in identifying input/output needs.

Users actively participate in developing test plans.

10. Vendor Participation

Long-term partnerships have been established with key vendors/consultants.

Vendors/consultant form an integral part of the systems delivery process.

11. Programmer/Analyst Empowerment

Team members participate in project planning.

Team members participate in decisions regarding resource allocation to projects.

Project schedules are determined in consultation with team members.

12. Product Quality

Users perceive that the system meets intended functional requirements.

The information provided by the systems meets user expectations.

Systems meet user expectations with respect to response time, flexibility and ease of use.

Users are satisfied with the overall quality of the systems.

13. Process Efficiency (items for this scale are reverse-coded)

Projects usually overrun budgeted costs.

Schedule overruns are common in most projects.

Backlog of development work is high.

Fixing bugs and other types of rework account for a significant proportion of systems development effort.

14. Process Maturity

Which of the following best describes the systems development process in your organization (check one).

Primitive: No formalized procedures or project plans exists. Very poor understanding exists of key process issues.

Repeatable: Can repeat tasks mastered in previous projects. Process dependent on accumulated experience of individuals. New tools and methods cannot be incorporated without risk, due to lack of a process framework.

Defined: The process is established and well understood. The process works in most normal and crisis situations. Sufficient data about process is not collected to analyze process efficiency.

Managed: The process is measured and controlled, metrics are meaningful and well defined. Systematic record of process performance measures is maintained.

Optimized: Systematic process improvement is done. Process performance is regularly monitored and used as a basis for improvement.