Systems Analysis and Design Models Revisited: A Case Study[1]

S.V.R. MADABHUSHI
University of Oklahoma

MARY C. JONES
Mississippi State University

R. LEON PRICE
University of Oklahoma

The development of information systems (IS) emerges from the need for more expressive designs that could work well in diverse situations. Over the years, several methodologies have been proposed that offer tangible alternatives that capture solutions to the problems in context, and to a limited degree for those that could potentially occur in time. This paper describes the advantages and pitfalls within each of the known approaches and the reasons why no single methodology has gained general acceptance. Efforts to integrate these approaches often requires one to determine the right mix of features (Lyytinen, 1987). A case study of a large manufacturing organization from the midwest is presented and a new methodology that attempts to blend in pertinent features of the existing models of IS design is proposed.

A survey of the literature indicates that information systems (IS) are subject to several deficiencies including lack of user involvement in the process; failure to incorporate cognitive, behavioral, and organizational issues into the design; poor user interfaces; etc... (Baroudi, Olson, & Ives, 1986; DeBrabander & Edstrom, 1977; Doll & Torkzadeh, 1990; Edstrom, 1977; Ein-Dor & Segev, 1978; Franz & Robey, 1986; Larcker & Lessig, 1980; Robey & Farrow, 1982; Robey, 1983; Zmud, 1979). In addition, Lyytinen (1988) points out that as many as one-half of the information systems developed are subject to severe problems that could be considered as failures, and that such failures may occur in the design/development process as well as in use and operations.

Following the framework by Ives, Hamilton, and Davis (1980), Lyytinen (1987) surveyed and categorized problems of the IS process. For example, six major problems in the IS development process are (1) ambiguous, narrow, and conflicting goals; (2) technology that restricts choices and is overly susceptible to change; (3) problems in economy due to inaccurate calculations and weak foundations; (4) process features encompassing poor communication, quality control, and domination by the systems analyst; (5) neglect of behavioral and organizational issues; and (6) a highly rationalistic self image (See also Anderson, 1989; Cheney & Dickson, 1982; DeBrabander & Thiers, 1984; Gorla, 1989; Huber, 1983;

Manuscript originally submitted October 16, 1991; Revised March 3, 1992 and June 8, 1992; Accepted June 22, 1992 for publication.

Jenkins, Naumann, & Wetherbe, 1984; Mahmood, 1990; Zviran, 1990). Of the six problems listed, further study by Lyytinen (1988) revealed that the problems pertaining to goals, process features, and organizational issues are the most common.

Also included in Lyytinen (1987) are five use and operations process related problems: (1) operations problems such as awkward interfaces, slow and unreliable design, and difficulty in usage; (2) data problems such as incorrect, irrelevant, incomprehensible, and missing data; (3) conceptual problems such as ambiguity and/or misunderstanding; (4) people related problems such as power shifts and negative impact on work; and (5) complexity in understanding, maintenance, and use (See also Alter, 1980; Cerullo, 1980; Shneiderman, 1981). Lyytinen (1988) adds that conceptual and data problems occur frequently.

There are several methodologies that facilitate the IS process and help alleviate its deficiencies. Among the more widely used are the Systems Development Life Cycle (SDLC) and Prototyping. Others include the Pragmatic Input/Output Constructive and Operational model (PIOCO) (Iivari and Koskela, 1987), the Evolutionary Design model (EDM) (Lucas, 1978), the Organizational Change model (OCM) (Alter & Ginzberg, 1978; Alter, 1980), the Bargaining model (Kubicek, 1983), and the Discourse model (Checkland, 1981; Lanzara, 1983; Lanzara & Mathiassen, 1985). McFarlan and McKenney (1983) observe that a given methodology may be more appropriate for a particular situation than another, and that the type of methodology chosen may impact the success of an information system. However, in practice, methodologies selected for use are often inconsistent with the given situation (Saarinen, 1990). Efforts to integrate the known methods require the determination of the right mix of features ideal for the given situation (Lyytinen, 1987). However, in reality methodologies are rarely integrated effectively (Saarinen, 1990).

The purpose of this paper is two-fold. First, we examine each methodology to determine its key advantages and pitfalls. Such a description is essential to assess whether the selected methodology addresses or is capable of addressing the aforementioned design/development and use/operations problems. Second, we attempt to integrate the existing methods by proposing a new methodology, and we examine the outcome of our proposed method by presenting a case study.

This paper is organized into five sections. The first section provides the conceptual background wherein critiques of each methodology are detailed. An outline of the existing system at the manufacturing company in this case study is provided in the second section. The third section deals with the proposed methodology and a discussion of its features. Section four provides the lessons learned, and the conclusions are provided in section five.

Conceptual Background

Classification of methodologies is a fascinating subject of study in its own right. A good classification provides better insight about the similarities and dissimilarities between methodologies. We adhere to the classification used by Lyytinen (1987) because of its clear approach. Thus, methodologies are classified as engineering, learning, or dialogue models. Engineering models are those that approach IS development from the perspective of the technical aspects of the system: these models include SDLC, prototyping, and PIOCO. Learning models are those that approach IS development as an individual and group learning process. Included in this category are evolutionary design and organizational change models. Dialogue models are those that approach IS development from the perspective of bargaining and inquiry. The bargaining and discourse models fall in this category. In the following section, we present critiques of several existing methodologies with supportive literature. For additional information, refer to Lyytinen (1987, 1988).

Engineering Models

System Development Life Cycle (SDLC). The SDLC is functionally designed assuming rational individual behavior and represents the system through a series of stages of refinement and transformation. These stages may be classified in several ways (see, for example, Awad, 1988; Hussain & Hussain, 1985; Lucas, 1986; Martin, 1991; Taggart, 1990; Zmud, 1983). Schemer (1987) groups the stages into four major classes that include (1) specification of problems, system and software requirements, and the conceptual design; (2) development of detailed design, coding, testing, and establishment of operational procedures; (3) implementation of acceptance tests, user training, and conversion; and (4) operation and maintenance.

Requirement specification is the most crucial phase in the life-cycle approach. Methodologies and techniques that provide requirement specifications include: (1) Structured Systems Analysis (SSA) (DeMarco, 1978; Gane & Sarson, 1979; Mendes, 1980);

(2) Structured Analysis and Design Techniques (SADT) (Ross, 1977, 1985); (3) Problem Statement Language/Problem Statement Analyzer (PSL/PSA) (Teichroew & Hershey, 1977); (4) Software Requirements Engineering Methodology (SREM) (Bell, Bixter, & Dyer, 1977); and (5) The Entity-Relationship Model (ER Model) (Chen, 1976). A life-cycle model offers high predictability, stability, control of the development process, and rational problem-solving behavior. However, it has several shortcomings including inflexibility, limited target scope, disregard for ambiguous context during design (Lanzara, 1983; Lehman, 1980; Lyytinen, 1987; Parnas & Clements, 1985), and sequential task flow (Shoval & Pliskin, 1988).

Prototyping. Because the life-cycle approach largely ignores the dynamic nature of systems, an alternative is the prototyping process in which requirements and functions are developed as the system is implemented (Appleton, 1983; McCracken & Jackson, 1981). It is based on the premise that "users can tell you what they want changed in an existing information system more easily than they can tell you what they want to be developed in (a new) information system..." (Necco, Gordon, & Tsai, 1987 p. 464).

Prototyping addresses the entire system, including the operations and development environments. This approach allows the user to refine and modify a rough working model of the system. Implementation must immediately follow the users' request, which often creates difficulties because users fail to fully comprehend their needs, even during design. Other limitations of prototyping include (1) reliance on quick turn-around time between a user request and implementation; (2) the requirement for management to effectively handle frequent changes to the system; (3) unclear and ambiguous goal specifications; (4) likelihood of irrelevant, incomprehensible and/or missing data; and (5) a tendency to accept the initial version as the final product (Lyytinen, 1987; Shoval & Pliskin, 1988; Saarinen, 1990).

Following Lyytinen's classification of methodologies, there are two major sources that dictate the complexity of an information system: the extent of the IS application domain and the built-in multidimensionality of the system. Most models of IS development handle the complexity problem by using (a) hierarchical decomposition to cope with the domain complexity whereby the number of levels in the hierarchy are determined during the application and (b) levels of abstraction to address social complexity whereby the number of levels are determined from the underlying theory. The PIOCO methodology incorporates these levels of abstraction as metamodels and is described below.

Pragmatic Input/Output Constructive and Operational Method (PIOCO). This model is based on the following five ideas (Iivari & Koskela, 1987):

> *Decision-making Orientation:* The design process is viewed as a sequence of transformations of successive representations of the process with the steering committee inquiry process supporting the decisions.
>
> *Contingency Approach:* The model integrates most of the IS design methodologies and techniques into a common framework to promote a flexible, situation-dependent framework application. This implies an underlying assumption that no detailed design methodology is suitable for all situations.
>
> *Balanced Organizational, Conceptual and Technical View:* The model views IS development as a form of planned and restricted organizational change. Its initial specification is independent of technical implementation.
>
> *Dynamics of The IS Design Process:* This approach involves three phases. The first, or main, phase defines the steps of each new life-cycle. The second phase, known as the learning phase, suggests a flexible mode of process planning. The third phase describes the steps involved in the main-phase dynamics.
>
> *Information System Assessment:* The model suggests that the IS design be effective, efficient, and meet user satisfaction levels. Effectiveness refers to the changes in the organizational environment due to IS development.

The PIOCO methodology is a refinement of its precursor, the Finnish model based on the Pragmatic Semantic and Constructive methodology (PSC). The latter, also a refinement of the life-cycle approach, is used to simultaneously process all abstract viewpoints during each phase of systems development by dividing decision-making into a tri-level hierarchy with sequential control at each level (Lyytinen, 1987). These levels are referred to as pragmatic (P), semantic (S), and constructive (C) levels.

The pragmatic model of an IS is defined as a restricted and planned change in the host system/organization. A change may affect factors such as automated or manual information systems, personnel, organizational arrangements, and working procedures. This level reflects the perspective of all interest groups affected by the organizational change. Therefore, a system's impact and purpose are studied at this level. In the PIOCO model, the P-metamodel is used to study cost versus effectiveness.

Input/output relations are studied at the semantic (S) level in the PSC model, and from the I/O metamodel of the PIOCO model. The purpose is to determine the primary information (data) and its processing rules independently of technical solutions and the user-system interaction. In the PIOCO model, this metamodel also involves an analysis of cost versus user satisfaction, thus, reflecting the user perspective.

The design and implementation of the system occurs at the constructive (C) level in the PSC model, and by using the C/O metamodel in the PIOCO methodology. The overall efficiency of the system can be analyzed at this level.

The principal advantage of the PIOCO model over the life-cycle design is the review of the impact of the system at the pragmatic level. Its design offers clarity through repetition, providing more adaptability than the life-cycle model. The PIOCO methodology incorporates corresponding description languages, a process model for IS design, and a model for choice and quality criteria (Iivari & Koskela, 1987).

Because tasks at the P-level are described as restricted and planned changes that form an investment entity, the model is oriented toward small scale development of individual or clustered application-oriented systems. In the case of PSC model, empirical evidence is often ambiguous with regard to the effects of changes in the IS context on IS problem areas such as ambiguous goal specification, technological restrictions, and organizational-view (Lyytinen, 1987). For example, using the P-level abstraction in the PSC methodology, one could potentially reduce or correct the problems in goal specification because of the steering committee's inquiry process. However, it may be erroneous to assume that using a steering committee reduces or eliminates the goal problems.

Learning Models

Evolutionary Design Methodology (EDM). The evolutionary design methodology stresses the role of organizational learning by structuring the IS development process in a hierarchical decomposition of activities. Learning is expected to take place during the user's learning cycle which is categorized into two categories — mature and immature phases (Lucas, 1978; Lyytinen, 1987). This methodology is primarily used for designing information systems that are primarily oriented toward decision support. Imprecise requirement specifications restrict the use of this method to small scale IS designs (Keen & Scott Morton, 1978). Disadvantages of this methodology include ambiguous and/or conflicting goals due to the user's learning cycle, problems due to poor quality of calculations, and improper foundations (Kling, 1980). In addition, problems pertaining to data and people (such as power shifts and job qualification changes) are reduced but not eliminated (Kling, 1980; Lyytinen, 1987).

Organizational Change Methodology (OCM). In this methodology, the IS development process is perceived in terms of organizational change strategies that developers can effectively use to improve the chances of successful IS implementation (Lyytinen, 1987). The principal focus of this method is related to the changes in the user and organizational environments. Ginzberg (1978) suggests two theories that monitor changes in both environments — the planned change model and the innovation model. The planned change model describes an organizational change in three stages (Lewin, 1952) as unfreezing, moving and refreezing. In contrast, the innovation model consists of a sequence of steps followed during the adoption of innovation. One example of the successful implementation of the latter approach is the ISAC methodology (Lundeberg, Goldkuhl, & Nilsson, 1981).

One striking advantage of the OCM is the built-in provision for learning new organizational behaviors — a feature found lacking in the evolutionary designs (Lyytinen, 1987). In this method an organization is treated as self-regulating, with an inherent ability to adapt to its environment through learning. The OCM is based on the assumption that changes are a function of an individual's attitude. However, this is one weakness of the OCM because an individual's behavior, in addition to his/her attitude, may cause or prevent the change.

Dialogue Models

Bargaining Methodology. Kubicek (1983) describes IS development as a mixed political conflict-cooperation game. It is assumed that the power of an organizational group is a function of both its skills and its bargaining position. Conflicts during IS development are resolved through bargaining and political maneuvering, but the methods are unclear. Implementation of this approach is generally found in European countries (Lyytinen, 1987).

Discourse Models. In this methodology, the IS development process employs argument and discourse to achieve rational actions, allowing diverse viewpoints to generate and interpret tasks (Lanzara & Mathiassen, 1985). The development and use processes are interwoven to create, share, maintain, and critique the existing scenario in an organization, whereby problems are defined for collective organizational action. In this methodology, there is no life-cycle of the output (Lyytinen, 1987). One example of implementation is the Soft-Systems Methodology (Checkland, 1981). Although prototyping and discourse methodologies are both commonly construed as group-based systems, in the former method, IS development is perceived from a technical/engineering standpoint, whereas in the latter approach, it is viewed as a social process.

Literature Summary

Understanding the IS development process is essential in determining the success of an information system. A survey of the literature reveals a variety of attempts to structure the development process in order to accommodate both rapid changes in IS environments and new types of information systems. It can be readily verified that each methodology plays a significant role in presenting a distinct perspective to IS development. For example, SDLC models can handle technical and operational problems effectively; prototyping may be appropriate when uncertainty about the functions of IS exist; and dialogue models may facilitate more effective goal setting. Table 1 illustrates the IS deficiencies and the methodologies that best address them.

Integration of these perspectives requires one to identify the right mix of features (Lyytinen, 1987). The PIOCO model is an attempt in this direction. This approach attempts to integrate prototyping and organizational change methodologies under the assumption that no single methodology suffices to address all situations. However, it is oriented toward small-scale development of individual or clustered application systems (e.g., Decision Support Systems) rather than toward the large-scale development of total Management Information Systems. The authors of PIOCO consider the latter to be supersystems that are collections of individual information systems (Iivari & Koskela, 1987). In addition, this approach prioritizes effectiveness, user-satisfaction, and system efficiency in the order of decreasing precedence. In the following section, we describe a request system used at the manufacturing company in question along with its problems.

Description of the Organization and the Existing System

IS Deficiency	SDLC	Prototyping	PIOCO	Evolutionary	Organizational Change	Bargaining	Discourse
Ambiguous Goals			X			X	X
Technological Restrictions	X						
Weak Economic Foundations			X				
Poor Communication		X	X	X			
Behavioral and Organizational Issues			X				
Awkward Interfaces	X	X	X				
Missing & Ambiguous Data							X
Wrong Problem Solved			X	X			
People Problems			X		X		
Maintenance & Usage	X						

Table 1: IS Deficiencies and the Methodologies That Best Address Them

In 1988, one of the authors of this study was asked to consult with a large midwestern computer manufacturer*, in which the IS organization was operated as an internal contractor where contact with the functional areas of the company was governed largely by formal requests and service agreements. IS had developed a reputation of delivering poor quality and unresponsive service. The Company's new Vice President of Information Services wanted to address this image by changing IS into an aggressive Value Added Vendor of Information Services. To this end, the Company transferred a little over two-thirds of its development staff into four Information Centers (ICs) that were physically located in user areas. The Company set new standard guidelines and used the ICs to produce Strategic Information Databases for their operating division by training the IC staff in the user tools that were to be supported. In a tactical sense, the IS objective was to solve IS problems with an inadequate system using a high degree of user interaction while simultaneously participating in the functional business decisions of the Company.

The key components in the organizational structure were (1) the four ICs, (2) Information Services Management (ISM), (3) Technical Services (TS), (4) the Application, Development, and Maintenance group (ADM), and (5) the end-users. The ICs were used as an initial interface between the end-users and the ADM group. IC personnel were to help locate the highest benefit opportunities within each division and recommend effective information services to support them. In addition, the ICs were used to expedite the relatively simpler needs of the end-users, thereby reducing the work overhead of the ADM group.

The ADM group was responsible for the analysis and development of relatively large new systems and the maintenance of old systems. This group interacted with end-users for IS problem specification and for system development and design. The Information Services group evaluated the allowable resources (e.g., an increase in the projected time to complete a project). The Technical Services group consisted of the Change Control Committee that determined whether designs approved by the ADM met existing standards.

Existing Request System. Figure 1 illustrates the existing request system. For simplicity, the analysis and design phases are not shown.

Typically the organization used service agreements and/or formal request systems to address the IS problems pertaining to the end-users. After the vice president of user management approved and forwarded a request to the ADM group, a clerk in the ADM group logged the request, numbered it, and assigned the job to a programmer analyst who then evaluated the projected time to complete the task. The request was then forwarded to one of the supervisors in the ADM group, who either responded with an initial "go ahead" if the projected time was acceptable or redirected it to the analyst for a better time estimate. After the initial go ahead, the request was submitted to the vice president of Information Services for a final go ahead. In the case of a "no go",

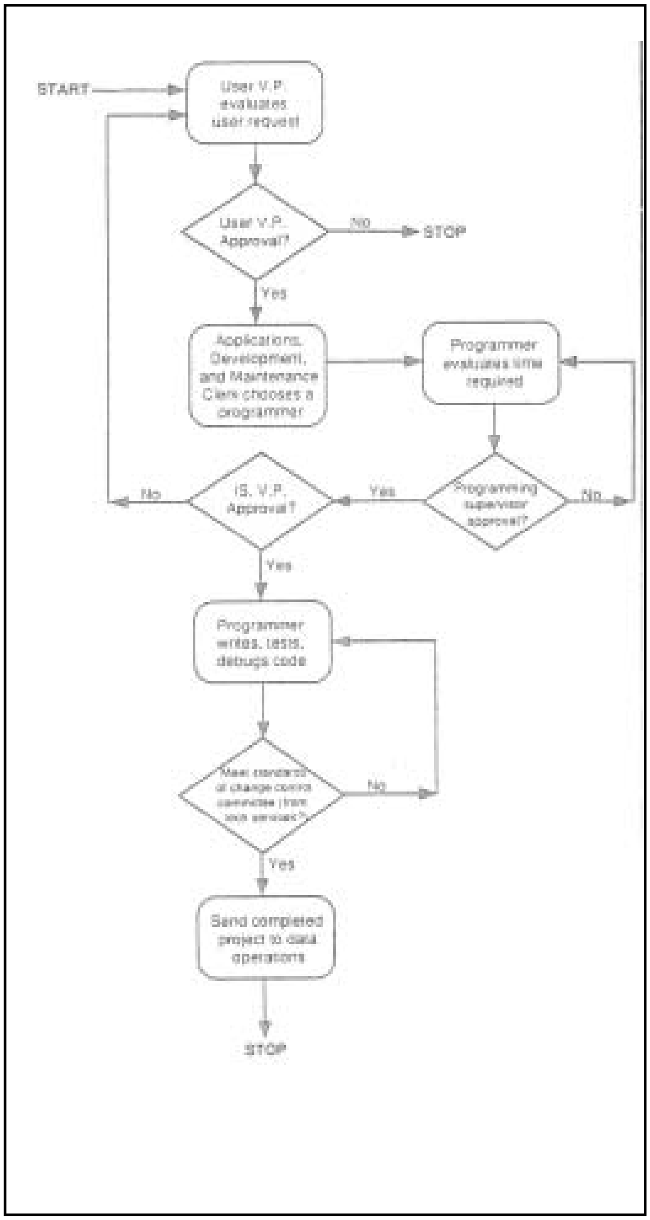

Figure 1: Current System

* Due to contractual obligations the name of the organization used in this study is not reported.

the request was turned down and end-user management was informed.

Upon a final go ahead, the request was rerouted to the ADM group where it awaited completion by the assigned programmer analyst. The analyst determined the requirement specifications by interacting with end-users. Thus, the design, development, and testing of any IS task was an outcome of a close interaction between the analyst and the end-user. However, prior to implementation, the task was forwarded to the Technical Services department where the Change Control committee checked whether the existing standards were met. If the committee decided otherwise, the job was returned to the ADM group for refinement. Finally, the task was forwarded to Data Operations for implementation.

Problems in The Existing System. An initial inspection of the existing system revealed that the underlying methodology for handling any formal requests or service obligations was a variation of the life-cycle approach. Thus, problems associated with this methodology, such as limited target scope, inflexibility, disregard for ambiguous context during design, sequential task flow, and avoidance of the dynamic nature of systems, could be readily felt.

Depending on the type of methodology chosen, for certain IS development and use process problems, Lyytinen (1987) maintains that there exists demonstrable evidence that there is a high likelihood of obtaining solutions for some problems through a change in the IS context component. However, he maintains that for other problems, evidence showing how the IS context component affects the solution potential is ambiguous. For example, if the development environment in the prototyping methodology uses application generators to capture the series of transformations needed, then the data problems are reduced because of user involvement. However, the conclusion that using application generators reduces or eliminates the data problems is erroneous: some IS process problems are unaffected by a change in the IS context component. In the existing system, all development and use process failures, excluding problems involving technology restrictions and those pertaining to operations, are either unaffected or only mildly reduced.

Although the idea of utilizing IC personnel to provide end-users with easy-to-use access to data extracted from the Strategic Information Databases and to respond to the users' simpler needs at the front-end is appealing, it is quite possible that both the end-users and the ADM group may receive mixed signals once they begin interacting. This may be one of the reasons that the Company reported a work backlog of five or more years.

Need For The Proposed System. There are several approaches to correct the deficiencies of the existing system. First, the existing system could be fine-tuned in order to reduce the backlog of work. For example, if systems maintenance is a part of the ICs' responsibilities, then the ADM group is tasked only with design and development work. However, this can lead to potential power conflicts and a major reshuffle in existing procedures.

Second, it is essential to realize that using any of the alternative methodologies to SDLC may yield only immediate solutions. The Company sought to make the IS department a Value Added Vendor of Information Services with a significant increase in end-user consulting. Because the systems developed at this Company are mostly user-driven, it is apparent that end-user satisfaction is an important asset to the Company's goals.

Considering an integrated approach, such as the PIOCO method, involves the use of a steering committee to review major IS decisions and a requirement that the IS design group must provide quality information to the committee for evaluating the effectiveness criteria. However, the manufacturing organization in question had no plans of creating a steering committee. Failure to do so may conflict with the Company's principal objectives. As a final alternative, the consulting team chose to fine-tune the system by blending pertinent features from the known models of IS design without major changes in the existing environment. The following section describes the proposed system and its features.

The Proposed Request System

The suggested model uses seven stages as shown in Figure 2. This section describes the tasks involved in each stage, followed by discussion of the merits of the consulting team's choices and experiences.

Stage 1. This stage includes (a) problem identification, (b) problem specification, (c) a rough outline of the technical structure of the system (e.g., in the case of individual information systems, this step involves the specification of input and output users, the corresponding information types, and their relationships to the users' activities), and (d) establishment of goals and overall deadlines.

As in the existing system, IC personnel deal with end-users to provide information and support needed using the Strategic Information Databases. Based on the

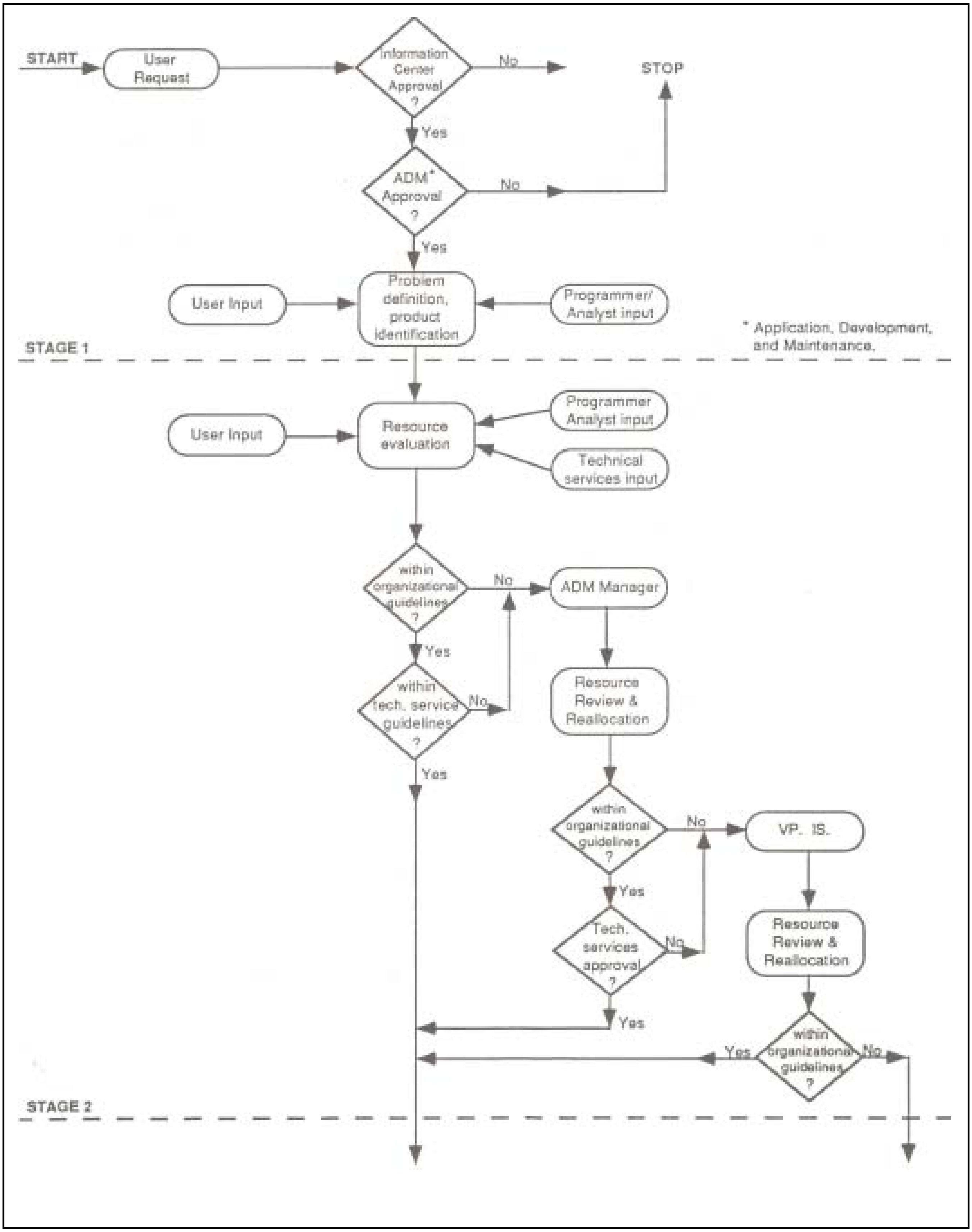

Figure 2: Proposed System

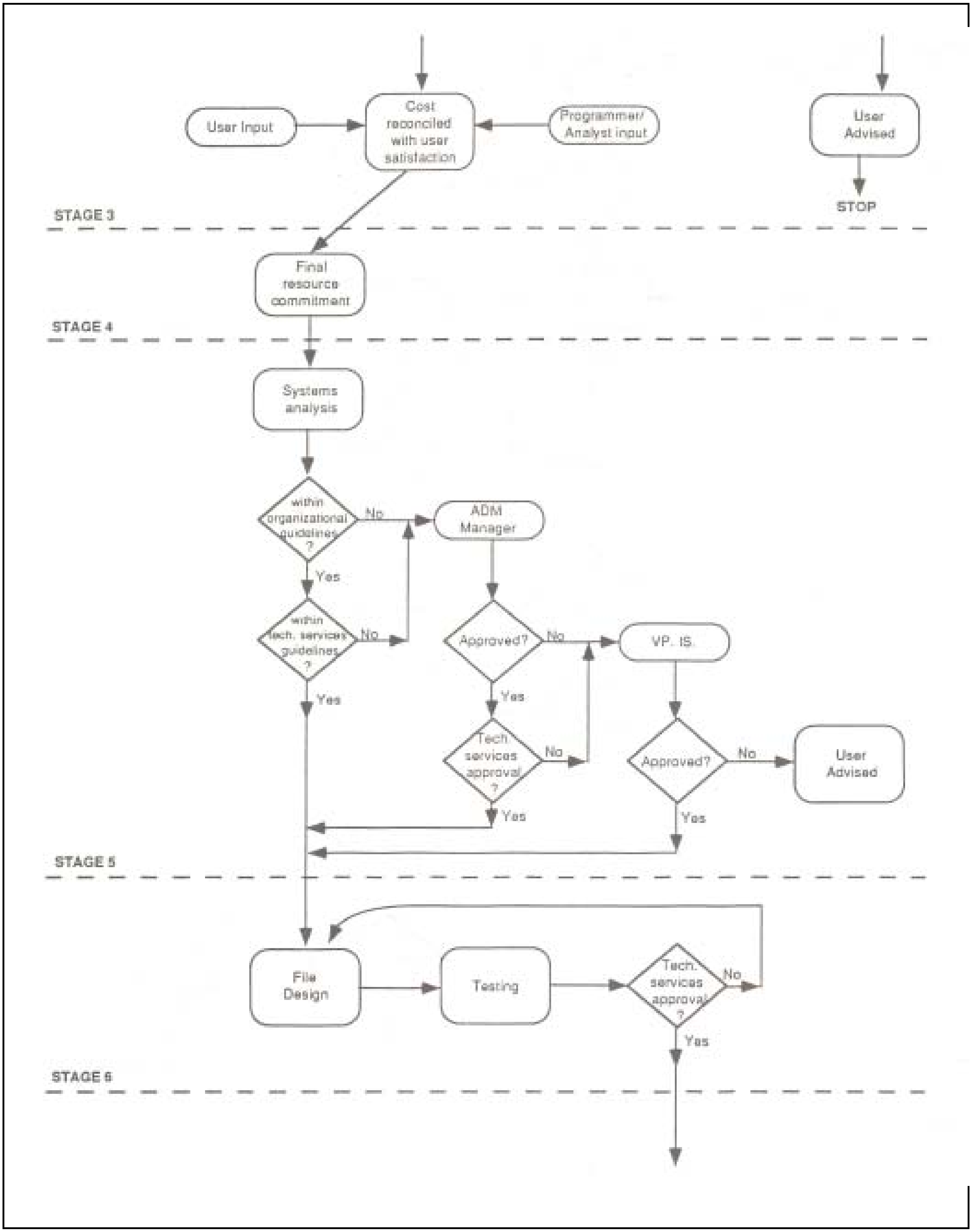

Figure 2: Proposed System (Contd.)

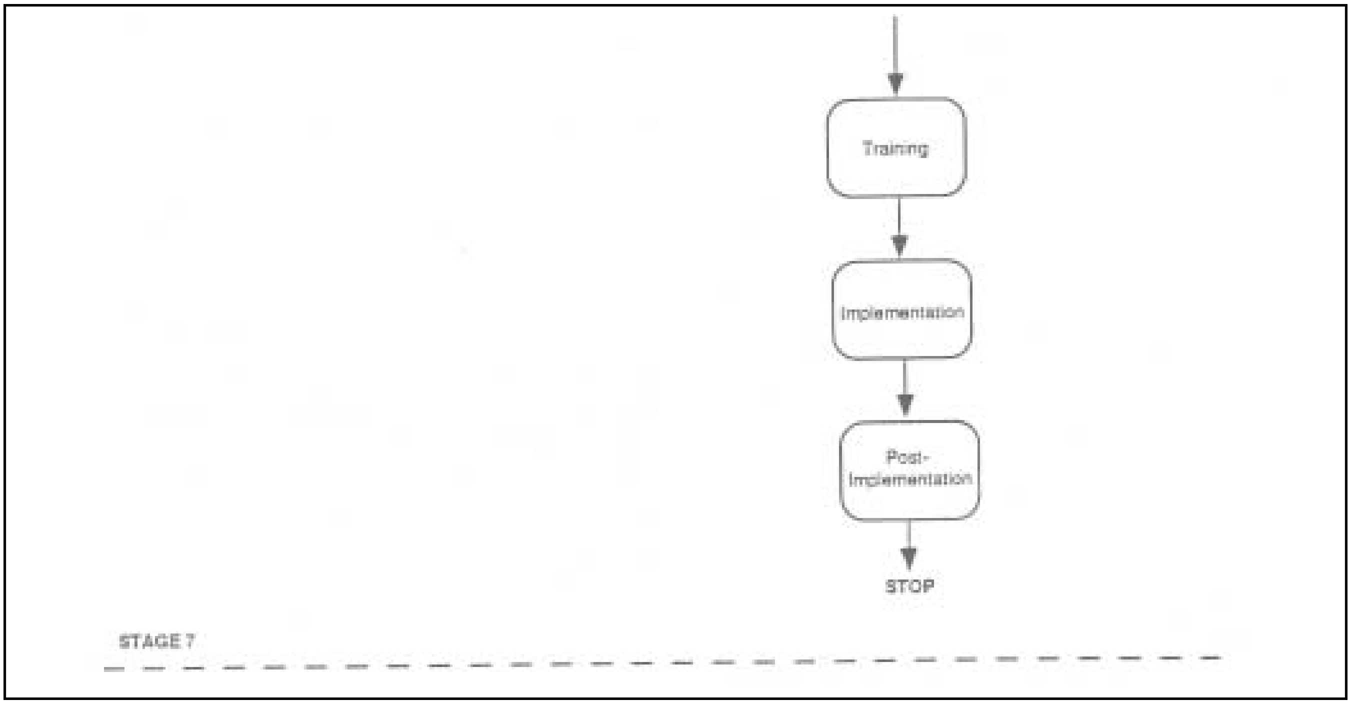

Figure 2: Proposed System (Contd.)

policy of the vice president of Information Services, the IC group may also cater to the relatively simpler needs of end-users. In contrast, if the IC group determines that a formal request is to be forwarded to the ADM group, they must interact with the end-users and the ADM group simultaneously for problem and goal specification, and for the establishment of a rough technical outline of the system.

Stage 2. This stage involves (a) resource evaluation/review, (b) cost-benefit analysis, and (c) feasibility analysis. Resource evaluation deals with establishing the estimated value of the programmer analysts' time requirement, the training time requirement, and the computer resources requirement. This must be accomplished by the ADM group. A cost-benefit statement must be developed by IC personnel and the ADM group in conjunction with end-users.

Because IS design generally includes explicit and implicit consideration of alternative information systems, the Change Control committee of Technical Services, the ADM group, and the end-users must agree upon and approve the solutions proposed for the IS problems in question as well as the specific methodology chosen for design and development. The Change Control committee mimics the role of a steering committee. Next, the feasibility of proposed solutions should be examined. Once the proposed solution is agreed upon, Technical Services must ensure that the system requirements for the initial study are well within the existing computer resources. Thus, the request-approval process also includes the Technical Services group.

If the needed resources are unavailable, a resource review must be conducted. Resource review involves a two step iteration process — a request for additional resources and re-examination and approval by the Technical Services group. The request for additional resources could be made at two levels of authority. First, it could be made to the manager of ADM for possible reallocation (see Figure 2). Personnel from Technical Services must then evaluate the new allocation to determine whether sufficient resources exist to support it. If resources are not sufficient, a request for additional resources must be made to the vice president of Information Services. The new allocation must again be examined and approved by the Technical Services group. Failure at this stage implies that the request must be turned down and that the end-user be informed.

Stage 3. This stage is also user-driven and includes preliminary specifications of (a) information (data), (b) information types (both syntax and semantics), (c) processing rules (ignoring technical specificities), (d) types of transactions, and (e) the nature of user-system interaction. The only groups participating in this stage are the ADM group and end-users. This stage defines the conceptual and infological specification of the information system.

Stage 4. It is during this stage that the IS area makes a final resource commitment to the project. Failure to do so implies that the project must be denied and hence involves debriefing of user management. This iteration may also require IS management involvement if total cost exceeds a prescribed operative level. Although not specifically shown in Figure 2, the final commitment to resources follows the same hierarchy as in Stage 2.

Stage 5. Given the preliminary specifications established during previous stages, the ADM group must develop a comprehensive and technically independent specification of the system during this stage. Data flow diagrams, data dictionaries, logic design, and input/output design are typical activities that must be specified. The impact of program and file design, security issues, and future maintenance requirements must be addressed in conjunction with end-user and Technical Services personnel.

Stage 6. During this stage, system design activities, such as file design, coding, and testing, should be accomplished within IS policy. The final design is approved or denied by Technical Services based on a cost versus efficiency analysis. If denied, the design is returned to ADM with recommendations for efficiency adjustments and must be revised for resubmission.

Stage 7. The analyst should work with the manager of systems training during this stage to ensure that the user is properly trained. Once training is completed, the product is formally turned over to Operations (Technical Services). The analysts must work closely with end-users as the system is implemented. Follow-up is a major responsibility and must be supervised by the ADM and the IC groups to ensure user satisfaction.

Features of the Proposed System. The proposed system is described in seven sequential stages with Stage 1 detailing problem specification. In Stage 2, the organizational context of the system is examined by the Change Control committee. Thus, Stages 1 and 2 reflect the view of all interest groups affected by the organizational change. In Stage 3, complexity and fuzziness of the preliminary information needs (assuming an error-free world and technological independence) are addressed. This implies the user's viewpoint of the information system. The final resource commitment in Stage 4 is yet another iteration process that ensures a systematic check of the costs and benefits. Stages 5 and 6 involve the design process and encompass a description of the system at the datalogical level, details of the technical implementation, and other supportive activities. User training and the actual implementation of the system are performed in Stage 7.

Notes From Our Experiences

The proposed methodology, when implemented, was shortlived because of the purchase of the Company by a competitor. However, there were good signs from follow-up interviews with the involved personnel. By the time the Company was purchased, the backlog had been reduced from five years to two years. With support from top management, the proposed methodology allowed for better communication within the Company and with end-users. The system appeared to be flexible because bending bureaucratic rules became commonplace. The IS area became responsive to measurement requirements and, more importantly, was perceived as a Value Added Vendor. In addition, target dates were met with better acceptance from the user. Table 2 summarizes activities and benefits at each stage in the new model. The new system is perceived to have several merits. First, there is no major revamping of existing procedures and no major changes in the organizational environment. This is an important asset for the Company, because each change can be perceived as a cost factor.

The new model also incorporates major features of existing IS design and development methodologies: for example, the P-level tasks in the PIOCO model could be established from Stages 1 and 2 of our model. The impact of IS development on the organizational environment is determinable during Stage 2 because all the stakeholders are present. The information for the decision process was obtained both from the built-in knowledge for decision making and the information at hand. Note that the principal difference between the proposed method and the PIOCO model is the absence of the steering committee. The decision making process in the approach is a collective and collaborative effort of the ICs, the ADM group, end-users, and Technical Services. Thus, problems of the IS development process such as conflicting goals, inaccurate calculations, weak foundations, poor communication, poor quality control, and neglect of organizational issues, are addressed effectively.

Finally, the new model is still perceived to be user-driven, and clarity and simplicity of IS design are obtained through repetition and review. Recalling the

Stage	Activities	Benefits
1	Problem identification Problem specification Outline of technical specifications Goal setting	Facilitates communication among IC group, users, and ADM Ensures that concerned parties agree on problem definition and system expectations
2	Resource evaluation and review Cost/benefits analysis Feasibility analysis	Establishes systems costs and benefits Facilitates realistic user expectations of costs and benefits Involves technical personnel to help make cost/benefit analysis more realistic
3	Specification of data, information type, processing rules, transactions and user-system interaction	Ensures users and ADM agree on general scope of system Establishes a blueprint for detailed systems design
4	Final resource commitment	Provides a means for aborting a cost prohibitive project
5	Detailed analysis of the system	Facilitates prioritization of system components Minimizes the focus on symptoms rather than problems in the current system
6	Detailed development Final review	Provides feedback on costs and technical issues before implementation
7	Training, implementation and follow-up	Ensures user understanding of system Provides feedback to ADM and technical services after system is in use

Table 2: Model Summary - Activities and Benefits of Each Stage

use process problems, it can be easily verified that during Stage 3, users address the complexity problem. The separation of infological and datalogical specifications to deal with primary and secondary data is an important step toward reducing data problems.

Conclusions

A comprehensive review of developmental methodologies is conducted. A new methodology that captures pertinent features of the methodologies reviewed is proposed, and a case study involving the proposed method is presented. Due to the purchase of the Company, much of the follow-up study is hindered. The proposed approach to structuring IS development must be perceived as an attempt to understand the mechanics of integration, rather than as a way that underscores the need for pragmatism. It is widely known that no single methodology suffices to address all the IS development and use process problems, and that there is an implicit and urgent need for pragmatism. However, from this experience, any attempt to integrate the known methodologies requires the determination of the nature of problems encountered in each methodology, and the frequency with which such problems occur. It is also essential to describe a framework that defines approaches to integration and case studies (such as the one here) showing demonstrable evidence of correcting the problems. Such an attempt may be a time consuming, yet worth-while, exercise. More research is needed to develop effective approaches to systems development.

References

Alter, S. (1980). *Decision Support Systems: Current Practice and Continuing Challenges.* Reading, Massachusetts: Addison-Wesley.

Alter, S. & Ginzberg, M. J. (1978). *Managing uncertainty in MIS implementation.* Sloan Management Review, 19, 23-31.

Anderson, E.E. (1989). The implementation of information systems for workers: A structural equation model. *Information and Management, 16*, 171-186.

Appleton, D. S. (1983). Data driven prototyping. *Datamation, 29*(11), 259-268.

Awad, E.M. (1988). *Management Information Systems Concepts.* Menlo Park, CA: Benjamin/Cummings Publishing Company, Inc.

Baroudi, J.J., Olson, M.H. & Ives, B. (1986). An empirical study of the impact of user involvement on system usage and information satisfaction. *Communications of the ACM, 29*(3), 232-238.

Bell, T., Bixter, D., and Dyer, M. (1977). An extendable approach to computer-aided software engineering. *IEEE Transactions on Software Engineering, 5E-3*(7), 6-15.

Cerullo, M.J. (1980). Information systems success factors. J*ournal of Systems Management*, December, 10-19.

Checkland, P. (1981). *Systems Thinking, Systems Practice.* Chichester, England: John Wiley & Sons.

Chen, P.P. (1976). The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems, 1*(1), 9-36.

Cheney, P.H. and Dickson, G.W. (1982). Organizational characteristics and information systems: An exploratory investigation. *Academy of Management Journal, 25*(1), 170-184.

DeBrabander, B. & Edstrom, A. (1977). Successful information system development projects. *Management Science, 24*(2), 191-199.

DeBrabander, B. & Thiers, G. (1984). Successful information system development in relation to situational factors which affect effective communication between MIS-users and EDP-specialists. *Management Science, 30*(2), 137-155.

DeMarco, T. (1978). *Structured System Analysis and Specification*, Yourdon, New York.

Doll, W.J. & Torkzadeh, G. (1990). The measurement of end-user software involvement. *OMEGA International Journal of Management Science, 18*(4), 399-406.

Edstrom, A. (1977). User influence and the success of MIS projects: a contingency approach. *Human Relations, 30*(7), 589-607.

Ein-Dor, P. & Segev, E. (1978). Organizational context and the success of management information systems. *Management Science, 24*(10), 1064-1077.

Franz, C.R. & Robey, D. (1986). Organizational context, user involvement, and the usefulness of information systems. *Decision Sciences, 17*, 329-356.

Gane, C. & Sarson, T. (1979). *Structured System Analysis,* Prentice Hall, Englewood Cliffs, NJ.

Ginzberg, M. J. (1978). Steps towards more effective implementation of MS and MIS. *Interfaces, 8*(3), 295-300.

Gorla, N. (1989). Identifying MIS research issues using a research framework. *Information and Management, 17*, 131-141.

Huber, G.P. (1983). Cognitive style as a basis for MIS and DSS designs: Much ado about nothing? *Management Science, 29*(5), 567-579.

Hussain, D. & Hussain, K.M. (1985). *Information Processing for Management.* 2nd ed. Homewood, IL: Irwin.

Iivari, J. & Koskela, E. (1987). The PIOCO model for information systems design. *MIS Quarterly, 11*(3), 401-419.

Ives, B., Hamilton, S. & Davis, G. (1980). A framework for research in computer based management information systems. *Management Science, 26*(9), 910-934.

Jackson, M. (1983). *Systems Development.* Engelwood Cliffs, NJ: Prentice Hall.

Jenkins, A.M., Naumann, J.D., & Wetherbe, J.C. (1984). Empirical investigation of systems development practices and results. *Information and Management*, 7, 73-82.

Keen, P. G. W. & Scott Morton, M. S. (1978). *Decision Support Systems: An Organizational Perspective.* Reading, Massachusetts: Addison-Wesley.

Kling, R. (1980). Social analyses of computing: Theoretical perspectives in recent empirical research. *ACM Computing Surveys, 12*(1), 61-110.

Kubicek, H. (1983). User participation in system design: Some questions about the structure and content arising from recent research from a trade union perspective. *Systems Design For, With and By the Users*, North-Holland, Amsterdam, 3-18.

Lanzara, G. F. (1983). The design process: Frames, metaphors, and games. *Systems Design For, With, and By the Users*, North-Holland, Amsterdam, 29-40.

Lanzara, G. F. & Mathiassen, L. (1985). Mapping situations within a system development project. *Information and Management, 8*(1), 3-20.

Larcker, D.F. & Lessig, V.P. (1980). Perceived usefulness of information: a psychometric examination. *Decision Sciences, 11*, 121-134.

Lehman M. M. (1980). Programs, life cycles, and laws of software evolution. *Proceedings of IEEE, 68,* 1060-1076.

Lewin, K. Group decision and social change. (1952). In *Readings in Social Psychology*, T.M. Newcomb and E.L. Hartley, Eds. 330-340, New York: Holt.

Lucas H. C. Jr. (1978). The evolution of an information system: From key-man to every man in person. *Sloan Management Review, 19*(2), 39-52.

Lucas, H.C. Jr. (1986). *Information System Concepts for Management.* 3rd ed. New York: McGraw Hill.

Lundeberg, M., Goldkuhl, G., and Nilsson, A. (1981). *Information Systems Development - A Systematic Approach.* Englewood Cliffs, NJ: Prentice-Hall.

Lyytinen, K. (1987). Different perspectives on information systems: Problems and solutions. *ACM Computing Surveys, 19*(1), 1-50.

Lyytinen, K. (1988). Expectation failure concept and systems analysts view of information system failures: results of an exploratory study. *Information and Management, 14*, 45-56.

Mahmood, M.A. (1987). Systems methods-A comparative investigation. *MIS Quarterly, 11*(3), 293-312.

Mahmood, M.A. (1990). Information systems implementation success: A causal analysis using the linear structural relations model. *Information Resources Management Journal*, Fall, 2-14.

Martin, M.P. (1991). *Analysis and Design of Business Information Systems.* New York: MacMillan Publishing Company.

McCracken, D. D., Jackson, M. A. (1981). *A minority dissenting position, Systems Analysis and Design: A Foundation for the 1980s*, North-Holland, Amsterdam, 187-195.

McFarlan, F.W. & McKenney, J.L. (1983). *Corporate Information Systems Management - The Issues Facing Senior Management.* Homewood, IL:Irwin.

Mendes, K.S. (1980). Structured System Analysis: A Technique to define business requirements. *Sloan Management Review*, 51-63.

Necco, C.E., Gordon, C.L. & Tsai, N.W. (1987). Systems analysis and design: Current practices. *MIS Quarterly, 11*(4), 461-478.

Parnas, D. L. & Clements, P.C. (1985). A rational design process: How and why to fake it, *Proceedings of the International Joint Conference on Theory and Practice of Software Development,* Springer-Verlag, Berlin.

Price R.L. & Madabhushi S.V.R. (1988) System analysis and design models revisited: A pragmatic application. *Proceedings of the 1988 Annual Decision Sciences Institute,* 642-644.

Robey, D. (1983). Cognitive style and DSS design: a comment on Huber's paper. *Management Science, 29*(5), 580-582.

Robey, D. & Farrow, D. (1982). User involvement in information system development: a conflict model and empirical test. *Management Science, 28*(1), 73-84.

Ross, D.T. (1977). Structured Analysis (SA): A language for communicating ideas. *IEEE Transactions on Software Engineering, SE-3*(1), 6-15.

Ross, D.T. (1985). Applications and Extensions of SADT. *Computer, 18*(4), 25-35.

Saarinen, T. (1990). System development methodology and project success. *Information and Management, 19,* 183-193.

Schemer, I. (1987). Systems analysis: A systemic analysis of a conceptual model. *Communications of the ACM, 30*(6), 506-512.

Shneiderman, B. (1981). *Software Psychology-Human Factors in Computer and Information Systems*, Cambridge, MA: Winthrop.

Shoval, P. & Pliskin, N. (1988). Structured prototyping: integrating prototyping into structured system development. *Information and Management, 14,* 19-30.

Taggart, W.M., Jr. (1980). *Information Systems: An Introduction to Computers in Organizations.* Boston: Allyn and Bacon, Inc.

Teichroew, D. & Hershey, E.A. (1977). PSL/PSA: A computer aided technique for structured system documentation and analysis of information processing systems. *IEEE Transactions on Software Engineering, 5E-3*(1), 41-46.

Zmud, R.W. (1979). Individual differences and MIS success: a review of the empirical literature. *Management Science, 25*(10), 966-979.

Zmud, R.W. (1983). *Information Systems in Organizations.* Glenview, IL: Scott, Foreman, and Co.

Zviran, M. (1990). Relationships between organizational and information systems objectives: Some empirical evidence. *Journal of Management Information Systems. 7*(1), 65-84.

[1] A previous version of this paper (Price & Madabhushi, 1988) appears in the Proceedings of the 1988 Annual Decisions Sciences Institute, pp. 642-644.

Seshu V.R. Madabhushi has authored or co-authored several articles in diverse fields including Computer Science, Management Information Systems, and Marketing. He has given papers at various national meetings. His research interests include IS design and development methodologies, AI applications, computer networks, parallel processing and computer architecture.

Mary C. Jones, Ph.D. is an Assistant Professor of Management and Information Systems at Mississippi State University . She is an active member in the Decision Sciences Institute and the American Production and Inventory Control Society. She has recently published articles in Datamation, Information & Management, and Computers & Security. Dr. Jones' current research interests are in systems development methodologies and information technology.

R. Leon Price is an Associate Professor of Management Information Systems at the University of Oklahoma. He is an award-winning professor in both teaching and research. Dr. Price has given papers in several national meetings and co-authored or authored over 50 articles, management reports and other publications. In addition, the students have consistently honored him as an outstanding lecturer throughout his career.