

EDP SYSTEM DEVELOPMENT METHODOLOGY:

AUDITABILITY AND CONTROL

Avi Rushinek and Sara Rushinek
University of Miami
School of Business Administration
Coral Gables, FL 33124/USA

ABSTRACT

This study is both a replication and an extension of prior work on system development. By expanding on the previous information with more updated information, a better description of the system development process is obtained.

First, the audit of system development is explained. Then two normative models of the process are described, pointing out the advantages of each. Finally, the major phases of the system development process are presented and the authors' personal views and experiences are expressed.

INTRODUCTION

System development is a relatively new method of dealing with the problems of an organization. As such, the development of good systems is still in the experimental stage. There are few formal rules to follow, and basic insight and experience still have an essential role in determining the quality of the resulting design. Therefore, the authors have found that what distinguishes a good system from a bad system is the success of the end result.

Each organization is unique, and a system must be specifically tailored to meet the organization's specific needs and desires. In this concept of an organization, the organization is regarded as a communication system closed on the users' (of the output) needs and wants. As an example, the first step is to assess the users' needs. Then products, processes, and distribution methods are planned so as to meet these desires effectively and efficiently. The planning information flows to operations when the plans are carried out and the product (goods or service) is produced. The product is then promoted so that users will be aware of its existence and how it will serve their needs. Finally, the product is distributed to them, at which time the needs and desires of users are filled. This closed-loop communication system should operate continuously if the organization is to be efficient. The example illustrates how the design of a system must constantly rely on feedback. As users' needs change, the changes should be discovered as quickly as possible by management, plans revised, and operations changed accordingly to best serve the revised needs (Litecky, 1981).

Auditing The System Development Process

An auditor may evaluate the system development process in two ways. One, as a member of the system development team (Hannye, 1977). Another, in an ex post review capacity when the system development process is evaluated. The two types of audit have different objectives and use different methods to gather evidence (Parker, 1977)

In the participation audit, the objectives are to ensure for a specific application system that controls are built into the system to safeguard assets, ensure data integrity, and achieve system effectiveness and efficiency. The primary method for the auditor to collect evidence is by observing the activities of the other members of the development team. This evidence is then evaluated against the auditor's model of the system development process.

In the review audit, the objectives are to reduce the extent of substantive testing needed for application systems and to make recommendations for improving the system development process. To collect evidence, the auditor uses interviews, observations, and a review of standards to obtain general and then detailed information on the system development process. He then evaluates this information to form a basis for hypothesizing strengths and weaknesses that may exist and to design compliance tests. Finally, the auditor selects a sample of application systems to determine whether the hypothesized strengths and weaknesses do actually exist (Davis, 1981).

Normative Models of the System Development Process

According to Caputo (1981), when auditing the system development process, the auditor must seek answers to two basic questions. First, do system design personnel perform all the activities necessary for the design and implementation of high-quality information systems? Second, are these activities performed well? In order to be able to answer these questions, the auditor needs a model to use as a basis for thinking about, evaluating, and approaching the audit of the system development life cycle approach and the socio-technical approach. The quality of the systems the auditors design depends on the appropriateness of the model used (Cerullo, 1981).

System Development Life Cycle Approach (SDLC)

The life cycle approach is the traditional method of systems development. The SDLC is a technique used to divide the system development process into a small number of distinct tasks with formal management control points placed between and during each phase. The objectives in using an SDLC technique are two-fold: to provide a proper and responsive communications channel among users, EDP auditors, hardware planning personnel, top management, and the data processing personnel responsible for developing the application systems.

Another method that is being used is to set up control points within each of the phases (Cerullo, 1981). During the development of the system, all requirements

specified at each control point must be satisfied before that phase or the next phase can be continued. Basically two types of control points are used in this SDLC: those affecting the quality of systems being developed from a computer processing point of view and those used to interface users and others outside the data processing department.

The life cycle approach arose out of the belief that a successful system development process relied on appropriate managerial and technical applications. A history of technical and managerial problems, such as cost overruns, inadequate economic evaluations, inadequate system design, management abdication, poor communication and inadequate direction brought about the development of the new approach. In those cases, the life cycle approach is the perfect solution to help overcome those problems. Each separate phase of the life cycle must be defined and goals set accordingly. The authors believe that careful planning and feedback to assure that the plans are followed is of utmost importance.

Sociotechnical Design Approach

Because the life cycle approach dealt exclusively with the technical aspects of a system, a new approach was developed to deal more effectively with behavioral problems that exist (Holley and Cash, 1981). Systems naturally degrade through lack of use, apathy, and sabotage. This approach arose as an answer to questions as to why behavioral problems occur and how they could be corrected.

The sociotechnical system is a process which optimizes the two systems: technological and social. The technological system's objective is task accomplishment while the social system's objective is to achieve a high quality of working life for the users of the system. Problems arise from neglecting the social system when implementing the new technical system.

The sociotechnical approach does not completely negate the importance of project management techniques and the traditional life cycle approach; rather, it makes use of those and adds to it the social approach. Thus, the sociotechnical approach is a more encompassing one, that believes that the life cycle approach is deficient when it comes to dealing with behavior problems. (Scott, 1979).

Evaluating the Major Phases in the System Development Process

A somewhat new approach has been advocated for the system development process. This approach, known as the normative model, includes the activities of the life cycle and sociotechnical approaches (Holley and Cash, 1981). There are ten major phases in the normative model of the system development process. They include 1) problem recognition, 2) management of the change process, 3) entry and feasibility assessment, 4) diagnosis and information analysis, 5) system design, 6) program development, 7) procedure and forms development, 8) acceptance testing, 9) conversion and 10) operation, maintenance and audit. If there are problems with the applica-

tion of any of these activities, the auditor is responsible for determining the effects of the quality of the application system produced (Dowell and Hall, 1981).

Management of the Change Process

Management of the change process is necessary throughout the entire system development process. This requires formal project controls and change facilitating aspects. The formal project controls may include budgeting, exception reporting and checkpoints.

Change facilitating aspects include unfreezing the organization, implementing the change, and then refreezing the organization.

Management proceeds to unfreeze the organization by preparing the organization for the change. They must inform the system users of the need for the change, the methodology of the change and what is expected from each of the users in terms of ability to adapt to the change. Feedback should be provided in this instance to the organization in terms of the user's attitudes and behaviors. Similarly, data processing managers must establish quality control procedures within their organization to ensure that they do it right the first time (Davis, 1981). Among several of the important aspects of getting the change implemented correctly are education, participatory decision making, and command.

Once the organization has been prepared for the change, the actual changeover takes place. At this time, the users and the organization are ready and should know what to expect in terms of the adjustment process. Once the change is complete, and the new system is functioning, the organization undergoes the refreezing process. Positive feedback on the new system ensures adaptation of the system by the system users. Refreezing prevents the system's users from resuming their old patterns of system usage. If the users are discovered to have their own private information systems, then the refreezing activities may not have proceeded properly and taken full effect.

Entry and Feasibility Assessment

The entire change process is begun when the system designer establishes entry into the organization. The entry phase is used to unfreeze the organization and to develop cooperation among the users. Group meetings would be an effective method to ensure proper entry. The entry phase is important because it is during this time that the systems designers help show the users the faults of the old system and the needs for the new one. Unless the designer is able to truly convince the users of the need for change, there cannot be a successful changeover. If this is the case, then the hopes and plans for a new system may have to be forgotten.

Once the system users are convinced of the need for the change, a feasibility evaluation of the new system is needed to ensure it will be productive and will evaluate the programmers' testing of the change, evaluate the validity of the change,

examine operating instructions, and generally review all program changes before final implementation in the production system (Stanford Research Institute, 1977). The four bases for evaluating feasibility under the normative model approach include technical feasibility, operational feasibility, economic feasibility and behavioral feasibility. There is a need to ensure that technology available is sufficient to support the proposed project. Also, with regard to the technology available, there must be the ability to acquire more and develop it sufficiently.

It is necessary to constantly review the feasibility of the system. New information is always gathered and, therefore, it must be reviewed. However, if the project is relatively small, feasibility analysis may be focused on at the conclusion of the development process. Feasibility may be difficult to determine in the beginning of the development process, since little is known about the system and the facts are still vague and difficult to determine. The objective of this entry and feasibility analysis to ensure the imposition of the new system upon the users was only advocated when the benefits outweigh the costs.

Diagnosis and Information Analysis

A proper diagnosis and information analysis is necessary to understand the social and technical systems currently existing and to develop the strategies for the new system. In order to have a basis for managing the change process, diagnosis is necessary. Strategies establish objectives and goals that help to point the design process in the right direction. Strategies may be developed through the information analysis.

In order to have an understanding of the existing conditions, the present organization is studied. The EDP auditor, user, and the project leader review the project organization, the arrangements with the user for communication, and the plans and work program for the design. This central point helps the project leader to establish a good working relationship with the user to ensure that the system reflects user requirements (Stanford Research Institute, 1977).

In addition, the designer must study the coordinating mechanism of the organization, and the willingness of the organization to change. Some of the major coordinating mechanisms advocated include: direction, organizational job design, selection, training, appraisal and developments, communication and control, and a reward system.

Also, strategic requirements need to be formulated, as they indicate the objectives and goals expected to be accomplished by the organization. Without the objectives and goals, there may be system failures that could have otherwise been avoided. Strategic requirements simply involve task requirements. They may be general or broad objectives or more specific. Once the strategic issues are evident, information systems can be designed to receive full benefits. There are many design types and by having knowledge of strategies, these designs may be evaluated effectively. Once information has been obtained on the previous system, the currently existing

system can be evaluated. Although there was already a determination of feasibility, a more detailed information package enables a more suitable institution of the proposed system. Before we can even start to consider systems development, we should understand the entire data processing spectrum and its short and long-range plans, and document an understanding of what presently takes place and what will take place (Davis, 1981).

System Design

System design has meant the design of the information processing system; however, this concept of system design is inadequate. The concept which this paper proposes is a full concept, one that involves both the design of the social and technical aspects of the system and the set of coordinating mechanisms. The methodology of the sociotechnical system design is called action research. The designer establishes a collaborative mode whereby users jointly share the responsibility for design functions instead of the system designer imposing a system design on users.

In the design of the system, the designer again needs some basis for thinking about the organization and the change process to be implemented (Scott and Booker, 1979). It is useful for the designer to think about the organization in terms of four major sets of interacting variables; task, technology, structure, and human-social. For example, the technical system design establishes relationships within and between the first three sets of variables; the social system design establishes relationships within the human-social set; the sociotechnical system design establishes relationships within and between the four sets of variables.

There are other relationships that are also important when considering the design process, such as environmental uncertainty, organizational structure, and the information processing capabilities of the organization.

Coordinating mechanisms existing between the social and technical systems focus on three things: job design, organization design, and control. Job design consists of the traditional model in which individuals are more interested in what they earn rather than what they do. The human relations model focuses more on the context in which the job is performed and the human resources model is self-directing and controlling. In organization design the designer sometimes faces the choice of selecting an appropriate structure for the users.

Information processing system design involves the design of the information flow, design of the data base, design of the decision support system, preparation of program specifications and preparation of hardware/software specifications. Developments in both the hardware and software systems require a continual updating and strengthening of the technical expertise of the auditor who works with his clients' EDP systems (Jancura and Nance, 1981). As such, the EDP auditor should have a wide spectrum of technical knowledge and competence.

If the system requires hardware and software not currently available in the in-

installations, hardware and software specifications must be prepared for the additional resources required.

Program Development

When the system design phase has been completed, the next phase in the system development process is program development. This involves a) designing; which entails the tasks of organizing the project, documenting the data base files, designing source documents in detail, and preparing program run writeups, b) flowcharting, c) coding, d) compiling, e) testing, and f) documenting. Generally the audit staff itself is involved in program development (Tashji, 1981).

Acceptance Testing

The purpose of acceptance testing is to identify as many errors and deficiencies in the system as possible prior to its implementation.

Normal processing attempts should also be made to try to crash the system to determine its tolerance to errors and ability to respond to exceptional circumstances.

According to the authors, it is very important from an auditor's viewpoint that the testing and documentation of test results be carefully planned and executed. One of the outcomes of the testing phase should be a test-bed of data for individual programs, and the overall system that should be properly recorded and maintained.

The conversion phase consists of organizing the project, conversion, monitoring system performance, and the final acceptance of system (Stanford Research Center, 1980). The conversion process may occur in one of three ways. First, the old system may stop totally and the new system takes over immediately. Second, both systems may run in parallel for a period, performing different functions, and with both outputs being utilized. Third, both the systems may run in parallel, performing the same functions, with the old system being used. Changeover to a new system will entail personnel training, installing new hardware and software, converting the files and programs, and scheduling operations and test running. The auditor is especially concerned with maintaining data integrity during the conversion process. There might be some tradeoffs between the integrity of data taken up on the system and the need to get the system running. When data is converted from one storage medium or one data structure to another, control totals must be developed to help identify any data corruption that occurs during the changeover process or any errors that exist in the data already.

During the operation of a system three types of changes may be needed: processing errors may be discovered that require correction; changes in the system environment may necessitate system modification; changes may be made to improve processing efficiency. The system should be reviewed on a regular basis by a team consisting of management, users, and auditors (Litecky, 1981).

Postaudit ensures that both the users as well as the organization have adapted

to the changes.

The controls and techniques governing the system development process are important because the adequacy and effectiveness of controls included in computer application systems are heavily affected by the methods and procedures used during the system development process. To ensure the accuracy, completeness, and validity of data being processed by complex computer applications systems, internal auditors must consider an entire system of controls that encompass data preparation, data entry, data communications, data processing and data distribution (Davis, 1980).

Personal Views and Experiences

The authors' personal views and experiences confirm the theoretical notions as they are described in this paper. However, in real life, they appear to be implemented simultaneously, without such a clear distinction among these approaches. For example, the normative model coexists frequently with the SDLC approach. We as auditors are deeply committed to the use of specific norms as prescribed by professional bodies and standards such as the American Institute of Certified Public Accountants (AICPA) and their Generally Accepted Auditing Principles (normative approach). Likewise, we find ourselves breaking a system's development into the smallest possible segments, simply because it is otherwise uncontrollable.

Although these approaches seem to be rather comprehensive, they underestimate the importance of the review of the software requirements. It is the opinion of the authors that these are the most notorious source of frustration, anxiety and numerous problems. The review should thus be included as a control checkpoint prior to hardware requirements. Many times it is the software that determines the hardware requirements. For example, certain software packages will run only on certain operating systems or on a limited number of systems. For instance, Lotus Development Corporation markets a spreadsheet called "LOTUS 1-2-3", which is configured to run only on IBM Personal Computers and its clones. In the event that a user must use this software, this automatically narrows the hardware options.

As a final note, the authors suggest the software required should be thoroughly investigated prior to considering hardware requirements as a major control break point. (See Figure 2 for an illustration).

Summary, Conclusions and Implications

This paper reports on two aspects of the system development process: 1) the different models that exist and 2) the major phases that are an integral part of all models.

The information presented on the two different normative models - the system development life cycle approach and the sociotechnical design approach - pointed out the benefits and problems of each. Then the ten phases were explained as they pertain to the process of system development.

The implications of this information as it relates to the auditor are signifi-

cant. The auditor must have an extensive knowledge of computer systems, as well as the ability to visualize the needs of the entity he is auditing. These are not only technical needs; social and behavioral needs are also very important.

The system development process is one that will always be upgraded and innovated on, based on the changing needs of the EDP auditor. Some of the others who have added to the models discussed include Dowell (1981), Hannye (1977), Holley (1981), Scott (1979), Barker (1977), and Tashji (1981). However, the articles that were written have gone beyond the scope of this text. Attention need be made to them as they further discuss the various other information available on accounting information systems development in EDP auditing.

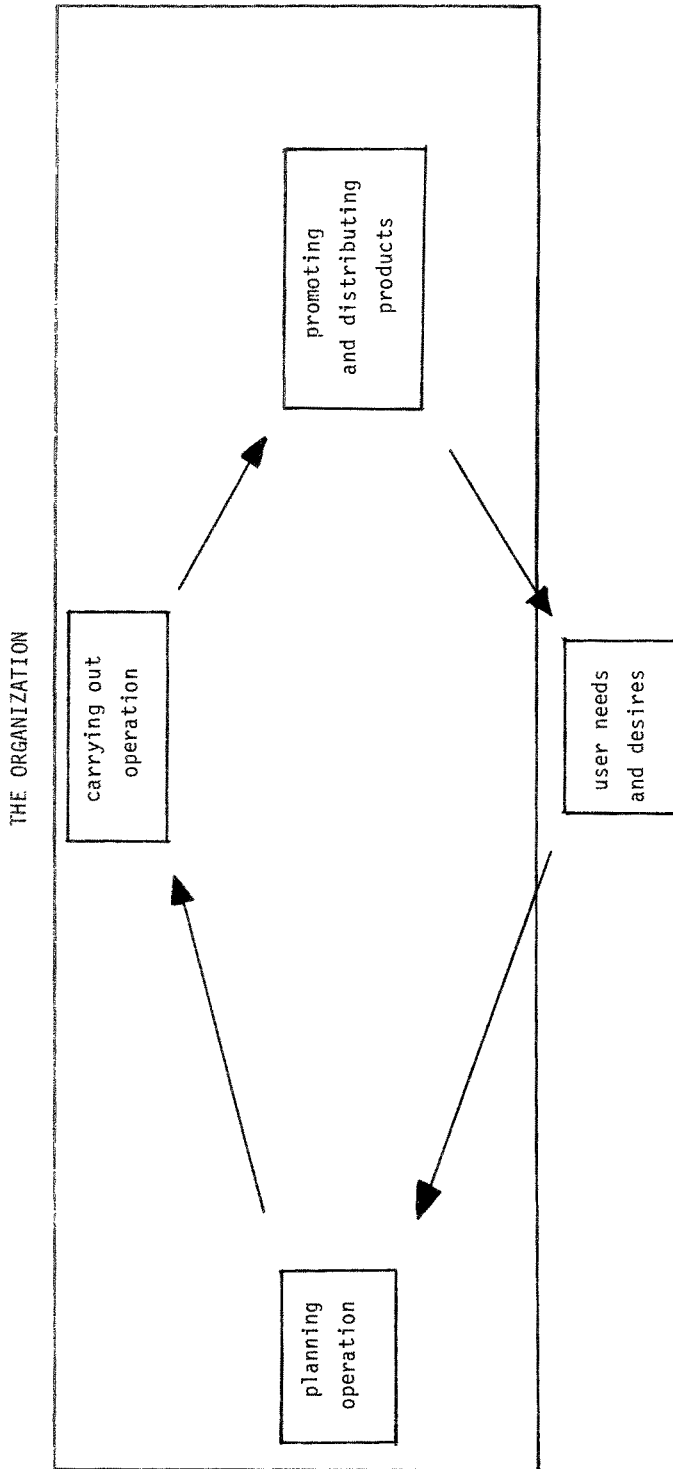


Figure 1. Systems Development Process

REQUIREMENTS OF:

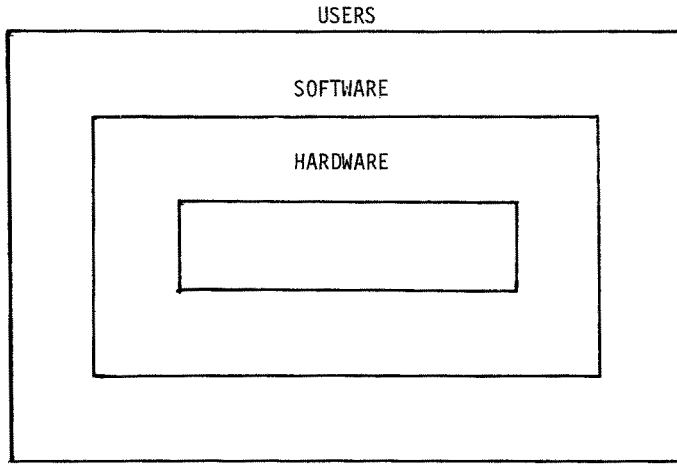


Figure 2. Control Checkpoint Relationships

References

- Caputo, Charles A., "Managing the EDP Audit Function, Part 1". The Internal Auditor, (February, 1981), pp. 73-77.
- Cerullo, Michael J., "Levels of Computer and Information Systems Knowledge Needed by Public Accountants". The National Public Accountant, (January, 1981), pp.19-23.
- Davis, Keagle W., "The Information Systems Auditor of the 1980's". Management Accounting, (March, 1981), pp. 40-47.
- Dowell, Dwayne C., and James Arthur Hall, "EDP Controls with Audit Cost Implications" Journal of Accounting, Auditing and Finance, (Fall, 1981), pp. 30-40.
- Hannye, George L., "Auditors and EDP'ers Benefit From Association in the Systems Development Process". Internal Auditor, (December, 1977), pp. 67-70.
- Holley, Charles L., and Daniel M. Cash, "Evaluations of EDP Systems Development". Journal of Systems Management, (June, 1981), pp. 16-21.
- Jancura, Elise G., and James J. Nance, "Proficiency Levels for EDP Auditors". Journal of Accountancy, (February, 1981), pp. 39-41.
- Litecky, Charles R., "Corporate Strategy and MIS Planning". Journal of Systems Management, (January, 1981), pp. 36-39.
- Parker, Robert G., "The Auditor's Role in Systems Development". Canadian Chartered Accountant, (September, 1977), pp. 56-58.
- Scott, Ronald L., and Jon A. Booker, "A Three-Phased Approach to the Systems Development Audit". Internal Auditor, (June, 1979), pp. 26-31.
- Stanford Research Institute, "Application System Development Controls". Systems Auditability and Control-Control Practices, (September, 1977), pp. 99-109.
- Stanford Research Institute, "System Development Life Cycle". Systems Auditability and Control-Audit Practices, (September, 1980), pp. 197-200.
- Tashji, Gabriel G., "The EDP Auditor's Role in System Application Development". Internal Auditor, (December, 1981), pp. 45-53.