PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° XX/AA

The Development Proposal: The First Step in Software System Construction

Arndt von Staa

Donald D. Cowan

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453-900

RIO DE JANEIRO - BRASIL

Monografias em Ciência da Computação, No. XX/AA
Editor: Prof. Carlos José Pereira de Lucena

ISSN: 0103-9741
Mês/Month, AAAA

The Development Proposal: The First Step in Software System Construction

Arndt von Staa[1]

Donald D. Cowan[2]

[2] Computer Systems Group, University of Waterloo

[1] arndt@inf.puc-rio.br, [2] dcowan@csg.uwaterloo.ca

Abstract

When developing software systems using the top-down design approach, it is extremely important to state first the objectives and success criteria of the development in a clear and comprehensive manner. Once these goals have been defined, the clarity, consistency, and breadth of this definition must be evaluated. The first document for software systems containing both the goals and their evaluation is the development proposal. In this paper we show how to construct a development proposal, explain why it should be constructed in this manner, illustrate its review, and evaluate its contents.

Originally appeared in: Staa, A.v.; Cowan, D.D.; "The Development Proposal: The first Step in Software Systems Construction"; *The Journal of Systems and Software* vol 1; New York: Elsevier; 1980; pp 107-120

1 Research supported in part by Financiadora de Projetos Especiais (FINEP), under contract CT-377.

2 Research supported in part by Canadian National Research Council Grant A2655.

In charge for publications:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22453-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br

Table of Contents

1 INTRODUCTION 1

2 RATIONALE FOR A DEVELOPMENT PROPOSAL 2

2.1 Main Objectives of the Development Proposal 2

2.2 Related Published Work 3

3 CONSTRUCTION AND REVIEW OF THE DEVELOPMENT PROPOSAL 4

4 THE REVIEW METHOD 5

4.1 Current-System Appraisal 6

4.2 Proposed System Appraisal 9

4.3 The Relationship of the Proposed System to Environment 13

4.4 Operational Feasibility of the Proposed System 14

4.5 Feasibility of Implementation 17

5 EPILOGUE 19

1 Introduction

The software development process normally follows a "stepwise refinement" pattern, and the steps are usually called phases. Several publications present a set of definitions for the phases and a description of the mechanics of documentation and verification for each phase and for the overall project. Some of these publications propose a "model-driven" development cycle [1, 2] where the results of each phase and their acceptance criteria are clearly defined. It is claimed that by following this development pattern, rather than other techniques, the number of retrofits is reduced and the quality of the resulting software is substantially improved.

In following a stepwise refinement or "top-down" development pattern the goals, functions, success criteria, and constraints of the proposed system should be stated first, so that there is a clear understanding of what is to be accomplished. This statement must be as precise as possible but must be a broad definition [3] since it is the first step in the refinement process.

We call this first document describing the software system the "development proposal" and believe it to be fundamental to the achievement of quality software products. This development proposal and its evaluation or acceptance criteria are the central issues of this paper.

In any top-down development pattern, acceptance of a step can be made only with regard to the knowledge which is available at the time the acceptance review is conducted. Thus it is quite possible that later steps in the refinement process will invalidate previously accepted items owing to increased knowledge or an evolving environment.

In the refinement process, a step may not be fully developed before proceeding to the next one. Such selective amplification of a portion of a step allows the partial implementation of a system to determine whether the system is likely to satisfy its goals and constraints. For example, such questions as the following are important to the system and should be answered early in its development:

- Are the performance criteria likely to be acceptable?
- Are the input and output formats acceptable to the users? and
- Are the operational constraints reasonable?

If the problems arise at this stage, the various steps in the design process can be modified before an extensive commitment is made.

The software life cycle consists of several phases, and the production of the development proposal is the first step. Several publications describe models for the partitioning of the whole development effort into phases. We shall not enter into more detail here, but refer the interested reader to other works [2, 4-6].

The remaining sections of this paper justify and describe the development proposal. Section 2 provides a rationale for the development proposal and briefly states how the concept evolved. In Section 3 an outline of the method for constructing and reviewing the development proposal is presented. Section 4 describes the review criteria used to evaluate the development proposal. Finally, section 5 reviews the entire concept of a development proposal. A detailed outline of the development proposal is presented in

the Appendix and can be used as a guide for the statement of the problem and its constraints.

2 Rationale for a development proposal

The successful development of a computerized system depends upon an early understanding of its goals, functions, success criteria, and constraints. This information must be acquired before any major planning effort is initiated. For example, there should be a clear understanding of the functions of the current system and the proposed replacement system, the constraints posed during its development and operation, and its interaction with other systems [1-4]. This background information will state major restrictions and mandatory utility and quality requirements and allow predictions and estimates. All of this information can be used as an aid for generating alternatives during subsequent phases [3].

2.1 Main Objectives of the Development Proposal

Software may be developed to add features to an already existing system, to replace some or all components of existing systems, or, finally, for an entirely new application.

In any of these cases, knowledge about the existing system and about the proposed replacement must be gained. It should be clear, though, that the degree of formalism should increase as the estimated development cost of the replacement system, or components, grows. Thus, for small improvements to an already existing system, a development proposal as described in this paper may prove to be an excessive effort. However, for major changes and developments of new systems, this effort should pay off in the form of a development subjected to far fewer surprises and crises.

The knowledge about the existing system is necessary in order to determine the feasibility of initiating the development, and to determine some of the goals which the addition or replacement system must satisfy. The existing system should be described by a brief but clear and precise document that highlights its known problem areas.

The proposed change or new system must also be described since the whole development effort will use this description as a departure point. At this initial stage, this description cannot be detailed; however, sufficient information must be presented to provide a clear, current picture of its goals, functions, operations, success criteria, and constraints. This description reflects the current state of knowledge about the system and may change substantially over the lifetime of the project. The proposed system is then compared to the existing system to determine whether the results are feasible and whether the proposed system covers the problem areas.

Although a comparison between the two systems is necessary, it is not sufficient to verify that the proposed system is a reasonable one. Many factors may create difficulties in the development or operation of the replacement system. Thus the environment for the creation and operation of the new system must also be known. This environment consists of development methods and constraints, operational procedures and constraints, and interactions with other systems.

As a final step, estimates must be shown for resources such as budget, manpower, and time, as these resources will be required to produce and install the new system. After gathering all this information one can attempt a cost analysis and hence determine whether the project is acceptable and should proceed.

The less the new development affects the existing system, the greater the precision to be expected in the estimates. Furthermore, as expected development costs decrease, so should the cost of the development proposal. However, even in the case of small projects or extensions, a development proposal is needed in order to state the objectives, the utility, and the quality requirements of the programs or system to be developed. If the proposal does not exist, we might end up with a patchwork that does not meet the objectives and requirement of the evolving system.

The development proposal should contain sufficient information for an economic analysis of the project. This analysis should not be incorporated into the development proposal since (1) it would introduce excessive detail (the analysis does not contribute sufficiently to the understanding of the proposal) and (2) the economic analysis should be reviewed after each development phase.

The development proposal consists of two major items: the proposal and a review of the proposal. The proposal simply states the problem and the current constraints on its solution. Thus the proposal contains several estimates and predictions and a tentative functional design for the new system.

The review provides a reasoned argument about the adequacy of the proposal. This argument guides the reader through the reasons for the decisions in the proposal and hence determines the level of knowledge at the time of proposal preparation. The review is not merely an acceptance criterion, but an explanation of the important portions of the proposal; thus it forms the heart of the proposal.

The development proposal described in this section should be viewed as a broad-brush problem description and not as a preliminary study or system specification. Both of these latter documents are to be produced after the development proposal has been accepted by management. Production of the proposal should not consume too many resources ; its main purpose is to serve as a commitment or contract between the users and system developers.

2.2 Related Published Work

The development proposal described in this paper is the result of an evolutionary process. During this evolution, development proposals were described and several new systems were formulated using each one as a model. Shortcomings were discovered and were eliminated.

The present development proposal is a minor evolution of another [7]. This one has been applied in an industrial environment to define a new medium-size information system. It has also been used as a "laboratory" in a systems design course. The major weak points found and corrected were precision of language and the necessity of environmental requirements (Section 4.3).

The current literature illustrates several different ways to produce development proposals. The method presented here has been proposed and used where it has been impossible to obtain a company-wide requirement analysis of the computerized system. The next few paragraphs mention briefly other methods described in the literature.

The method proposed by IBM's Business System Planning [8] requires the participation of people at the highest management level, including the president of the company. The objective of the business system planning method is to design an integrated information system. Such a development is not always possible, owing to the size, na-

ture, and geographical dispersion of the company. The business system planning method is very formalized and includes planning for the requirements analysis, determining typical manpower, office organization, and interview sessions. The results of applying this technique provide a solid understanding of the company's function and information needs without too much reliance on the present company structure.

PSL/PSA [9] proposes a documentation system and a formal mechanized analysis and review system. It is most useful during analysis in determining functional specifications, problem statement, and design. Although the development proposal and functional specifications are very similar on the surface, they differ immensely in detail. The development proposal is a sketch of the whole system and its constraints. On the other hand, the functional specifications must be precise, complete, and detailed.

Several other publications [2-5] mention the need for a development proposal and give some details about its contents. The January 1977 issue of IEEE Transactions on Software Engineering presented several articles concerning requirement analysis. Again the main concern seemed to be to produce quite detailed descriptions of what the system should do, i.e., a document close to a functional specification. However, no publication known to the authors presents a detailed description of a development proposal including its construction, contents, and review methodology.

3 Construction and review of the development proposal

Once the need for a new system or an extension has been observed, the first version of the development proposal should be produced, using the detailed guidelines presented in the Appendix. The first version of the proposal may not contain enough information to answer adequately to each item in the guidelines, so production of a final version is likely to be an iterative process.

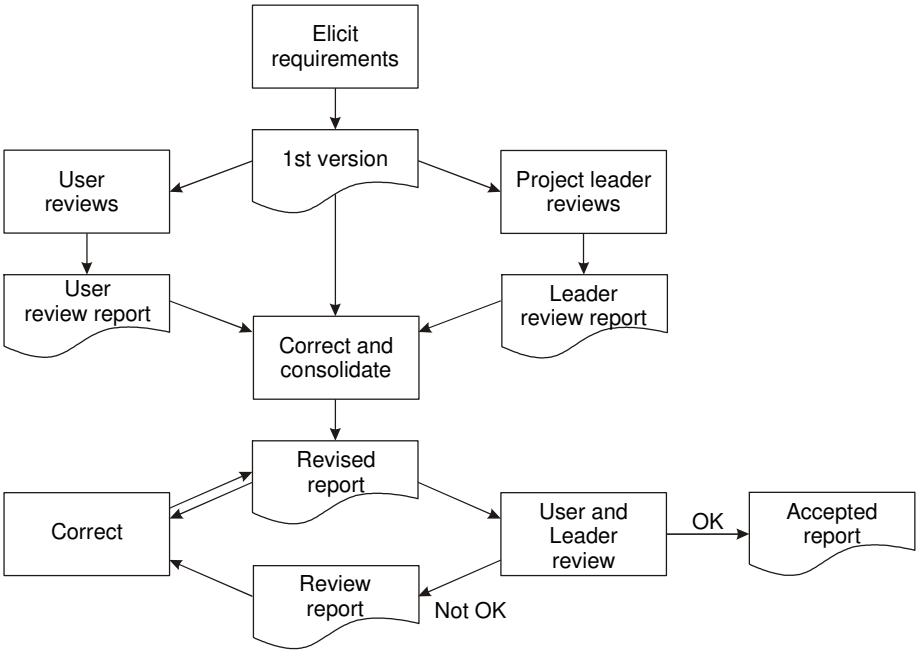

Figure 1. Construction and review process.

Once the first version of a development proposal has been produced, it is reviewed by the project leader (system designers, system architects responsible for the project)

and by the users or their representatives, and a review report is produced. Using the review report and the development proposal, it is the project leader's responsibility to ensure that the development proposal is complete and consistent and that there is general agreement about its contents.

Once the corrections have been made, the revised development proposal is reviewed again in joint sessions between the users and the project leader. This process continues until there are no more changes to be made and both users and the project leader agree that the development proposal is accurate with respect to the knowledge at the time the final version is produced.

At this point the development proposal is complete and both parts of the proposal and review are made available.

In Figure 1 we show steps in the construction and review process for a development proposal.

In some cases, the construction and review cycle for the development proposal may never converge to an acceptable version. Such an instability can occur for several different reasons. For example:

1. The user and project leader cannot agree on a final proposed system because the users are unable to specify the requirements of the system.
2. The corrections to the development proposal negate previous corrections since no adequate record-keeping mechanism for changes has been instituted.
3. The use of excessive detail prevents completion of the proposal since the users and project leader do not understand its purpose.
4. The system is developed independently of its final use. This is often the situation with systems developed as status symbols.
5. Neither the users nor the project leader actively participate in the construction or review of the proposal. Such a situation arises when one or both are technically immature and a thorough discussion is not possible.

There may be other reasons for lack of convergence of the versions of the proposal, but the examples are representative of most of the difficulties one is likely to encounter.

4 The review method

The main purpose of the development proposal is to serve as a base for the development of the proposed system. The proposal consists of two parts: a description of the current and proposed systems, and a review or evaluation of the adequacy of the current and proposed systems.

The description method has been presented in a previous section. The group preparing the development proposal amplifies each section and subsection of the outline given in the Appendix.

The review or evaluation method is presented in this section. As an evaluative procedure, the review must answer questions about the original description of the system to determine whether the description is complete. The answers to these questions often pinpoint deficiencies in the current description and hence are used to improve it. This process is an iterative one and several iterations may be required before all questions

are answered in a satisfactory manner. The final development proposal can then be used as a control standard during later stages of the system development.

Since each project is different, it is difficult to formulate a set of questions that covers every possible contingency. It is more reasonable to establish a set of standards or appraisal criteria against which the description can be measured. If chosen correctly, these appraisal criteria should cover the entire set of questions and in fact also anticipate any future questions.

This section presents a set of appraisal criteria that attempts to encompass all the probable questions. Different appraisal criteria may be based on identical data contained in the development proposal; in that case the data is interpreted in a different way. The appraisal criteria are grouped into five main classes:

1. the current system,
2. the proposed system,
3. the relation of the proposed system to the environment,
4. the operational feasibility of the proposed system, and
5. the feasibility of developing the proposed system.

Each of these criteria is further subdivided, and the subdivisions are described in the section corresponding to each class.

Several of the appraisal criteria overlap and hence are somewhat redundant. Redundancy is a desirable property since it provides a means for checking results and, hence, an informal error-detection mechanism. Of course, such redundancy may also be a source of inconsistency and hence cause errors. Thus evaluation using the appraisal criteria must be performed with caution.

4.1 Current-System Appraisal

This appraisal class provides a framework so that the understanding of the current system can be evaluated. This class is necessary since it is the first step in the justification of the development of a replacement system. If no current system exists, this appraisal class is irrelevant. In making the appraisal of the current system it should be noted that a lack of precision in any of the descriptions is very dangerous since the replacement system proposed may lack certain essential functions and thus cause wrong estimates and restrictions to be established.

The appraisal criteria for the current system may be subdivided into description, current organization, necessity, complexity, and security.

Description of the current system. This appraisal criterion is used to obtain an understanding of the functions of the current system. The appraisal should be made at a broad level and little or no analysis should be performed. A more precise and detailed understanding will be gained later during the analysis or functional specification phase.

Evaluation requires information about the following:

- the tasks performed by the current system, both manual and computerized;
- the goals of these tasks;
- the data flow between tasks;

- the service requests that may occur, and which tasks are triggered by these requests;
- the origin of the service requests;
- the periodicity, or volume per time unit, of service requests;
- the results required for each type of service request; the external inputs per task;
- the data maintained by the current system;
- the storage media, such as ledgers, file cabinets, tapes and disks;
- the expected and real elapsed time from service-request reception to output distribution; and
- the number of service requests that are not satisfied,
- and the main reasons for this lack of satisfaction.

Current organization. This appraisal criterion examines the current organization of the enterprise and whether it is adequate to operate the present system. This criterion is important since it provides an indication of whether a change to the organization is necessary or desirable. In many cases the observed need for a new system stems from inadequate organization. Installing a computerized system in such a case will not alleviate these organizational difficulties and, in fact, may even emphasize them. As a consequence the whole development effort may be wasted because of an inadequate product.

Replacement systems are frequently close in concept to the current system. Hence, weak spots in the present organization and operational environment should be well understood since they may be reflected in the replacement system and expected benefits will not occur.

Evaluation of the organization requires investigation of the following:

- the tasks performed by the present system; the data flow between tasks;
- the controls that are imposed on the operation of the present system;
- the operational difficulties of the present system;
- the volume of stored data;
- the storage media used for, the data; any access and retrieval difficulties;
- the number of operational users involved in the operation of a task;
- the number and complexity of processing steps for each service request;
- the number and organization (position and responsibilities) of persons (operative users, direct users, and indirect users) involved in each service request-here we should count actual persons not person-hours;
- the number of service requests that have not been properly processed and the reasons;
- the real and expected elapsed time between service requests arriving and output distribution;
- the number, size (number of data items), and complexity of forms used by the present system;
- the operating cost of the present system; the volume of service requests; and
- examples of each class of user, emphasizing training and skills.

Need for the current system. This criterion measures the importance of the present system to the user. Only the needs of the user for the current system are examined; the cost and operational difficulties are not considered.

Using this criterion, an attempt is made to determine the utility and value of the current-system features to both the direct and indirect users, since in many situations inadequate or unnecessary outputs or reports are produced. Quite often such outputs are found to be the key reason for upgrading a system that really is operating in a satisfactory manner.

Current-system complexity. The evaluation requires information about various factors:

- the tasks performed by the current system; the data flow in the current system;
- the effect of data that is imprecise, incomplete, and not current on the tasks of the system;
- the difficulty of performing each task-specifically, the volume of data accessed, the size of the output, the complexity of the operations performed, such as aggregation, sorting, association, computation, and the interdependence of the stored data, should be examined;
- the number of forms required; the volume of services required;
- the expected and real elapsed time from service-request arrival to output distribution;
- the number of steps required to produce desired outputs per service request; and
- the operative user-the training and skills required.

Security of the current system. This appraisal criterion determines the capacity of the current system to operate in adverse or hostile environments.

All systems whether they are computerized or not should be able to survive catastrophes, such as fires and floods, operational errors, such as input errors or oversights, and intentional errors or misuses of the system. This criterion should determine security and weaknesses of the present system and determine what would be acceptable security standards for the present system.

The evaluation requires information about the following:

- the ability of a task to function with data that is incomplete, imprecise, or not current;
- the degree of confidentiality of the data (what happens if some data is known to unauthorized personnel?);
- the direct and indirect users' views of the importance of the present system;
- the volume of stored data;
- the storage media used for maintaining data;
- the existing catastrophe protection measures;
- the existing access authorization measures;
- the difficulty of reconstructing destroyed data;
- the frequency with which data is incomplete, imprecise, or not current; and
- the number and organizational structure of the operative staff.

4.2 Proposed System Appraisal

In this section of the development proposal, the proposed computerized system is examined. The main items to be considered are a tentative design document, the impact of the proposed system on the user, and the expected benefits. One should also determine whether there are any requirements being placed on the proposed system which are either superfluous or unrealistic.

The appraisal criteria for the proposed system may be subdivided as follows:

- definition of the proposed system,
- definite restrictions on the proposed system;
- the life expectancy of the proposed system;
- the complexity of the proposed system,
- the complexity of the solution to the problem relative to the task to be performed,
- security of the proposed system, and
- benefits of the proposed system.,

Definition of the proposed system. This criterion allows the construction of a rough approximation to the functional specification and a subsequent evaluation of that specification.

Although at this stage there is not necessarily enough information about the system, an experienced designer should be able to produce an intuitive functional specification that shows how the proposed system will operate. Obviously, this preliminary concept is likely to be quite different from the system design that will eventually be produced; however, this preliminary design is necessary to estimate quantities such as costs, deadlines, and resource requirements. Furthermore, this approximate specification is necessary to gain an understanding of the main inputs, outputs, data bases, and processes that will have to be designed [3]. It should be stressed that this approximation is just that: an approximation, and not law. However, changes to this approximation must be justified so that we do eventually converge to a workable design.

The evaluation of the proposed system requires the following:

- a statement of the automatic and manual processes contained within the proposed system;
- the description of these processes; the data flow among the processes;
- the necessary data to activate each individual process;
- the output for each process;
- the data maintained by the processes-specifically the type of data but not the implementation details;
- the service requests and their descriptions; the origin of service requests;
- the frequency, or volume per time unit, of each service request;
- the expected elapsed time or response time from service-request arrival to output distribution;
- the responsibility for input data;
- the responsibility for storing and maintaining data; the need for the outputs;
- the security requirements for input data;
- the security requirements for output data;

- the security requirements for the data base (in all this description, the name "data bases" is used in the general sense of data, and includes data stored as on-line files, dismountable files, filing cabinets, and ledgers); and
- the expected volumes of inputs and outputs, data bases, and service requests.

Definite restrictions. This criterion determines the validity of any preimposed design and implementation alternatives. Preimposed design and implementation restrictions may be caused by certain standards or peculiarities of the site(s) of operation; thus languages, data-base systems, computers, input/output equipment, and other things may be predefined and cannot be freely chosen during design and implementation. Since such standards or preimposed alternatives are frequently based on current conditions and may change in the future, they should be reviewed and restated explicitly for each new project. The appraisal should be performed both overall and on an item-by-item basis. Since the restrictions vary from project to project, it is not generally possible to determine elementary items in advance. However, the form of the restrictions should obey the following pattern: (1) restriction description, (2) justification for the restriction.

Typical restrictive items are hardware, portability, operating systems, software support systems, languages, input/output equipment, and forms, as well as labels, density, and layout for each file, development procedures, documentation procedures, programming style, system types available (such as online systems, centralized systems, and networks), quality and maintenance requirements, and user interview restrictions.

There are several other possible restrictions that could be discussed under this appraisal criterion, but it is virtually impossible to produce an exhaustive list in a limited space.

Proposed organization. This criterion measures the impact a proposed computer system will have on the user organization and evaluates whether the user organization should be modified.

The success of a computerized system depends not only on the design and coding quality but also on the ability of the user to make effective use of the system. The responsibility of the user in both the areas of implementation and operation of the system must initially be made clear.

The evaluation of the proposed organization should consider:

- the processes provided by the proposed system; the data flow;
- the groups responsible for producing input data;
- the group responsible for storing and maintaining data;
- the outputs of the system and the destination of these outputs;
- the expected use to be made of these outputs; the storage media for the data;
- the mechanisms to gather and transcribe data into machine-readable form;
- the number and skills of all operations personnel; the number and skills of all maintenance personnel;
- the controls of the system, including security and quality control;
- the expected elapsed time between input and receipt of output-the so-called response time;
- any equipment restrictions; any software restrictions; and programming restrictions.

Life expectancy. This criterion measures the expected lifetime of the proposed computer system. As time progresses, the data and processing volumes for any system might increase substantially and hence saturate or exceed the system capacity. This expansion may occur because more data is being processed or more functions are being performed by the system. It is necessary to estimate a system capacity so that the life expectancy of the system is sufficiently large to justify the investment.

By estimating life expectancy it becomes possible to decide whether a new system should be developed, or the old system enhanced, before a crisis. Furthermore, it is possible to enforce expansion criteria at the beginning of a system and thus hopefully reduce costs of unexpected expansion.

An evaluation of life expectancy should consider:

- the expansion policy of the user entity;
- the integration and centralization policies of the user entity;
- the expected growth of the data base; the expected growth of service requests;
- the expected growth of future additions to the system; the current volume of the data base; and the current volume of the service requests.

Complexity of the proposed system. This criterion measures the operational difficulty that may be caused by the complexity of the proposed system. Operational difficulties with a computer system are usually caused by the design, and their elimination may induce larger development cost. Since the design at this point is only a first approximation, some of the operational difficulties will be real, whereas others will be caused by the naïveté of the system design. Stating the operational difficulties acts as a design control and is also an indicator of the necessity of amore careful analysis and system design to be performed during later phases.

An evaluation of complexity should consider:

- the processes involved;
- the data flow;
- the origin and responsibilities for data;
- the responsibility for storing and maintaining data;
- the data-gathering and data-transcription procedures;
- the controls;
- the output distribution;
- the volume of service requests;
- the expected maximum elapsed time between service-request arrival and output distribution;
- the estimated volume of processing per service request;
- the number of steps required to handle a service request;
- the ability of the system to handle data that is imprecise, incomplete, and not current;
- the storage media for data; security against accidents;
- security against corruption such as the misuse or willful destruction of data or programs; and general quality requirements.

Simplicity of solution. This criterion measures whether certain functions, outputs, and service requests are really necessary. Its main purpose is to verify whether the pro-

posed design can be justified and whether it contains superfluous requirements or false expectations. The evaluation of the simplicity of the solution requires the same items as in the subsection on current-system complexity (in Section 4.1). One should also include the expected use of outputs and expected users of system output.

Security of the proposed system. This criterion measures the ability of the proposed system to withstand human errors, machine errors, accidents, and corruption. It is sometimes called "robustness". Corruption is usually defined as the unauthorized use of data or programs or unauthorized changes to data or programs where such changes or uses are deliberate.

Security measures must be part of the system from the initial design stages. Some of these security measures are non computerized tasks that must operate harmoniously with the rest of the system. For example, output distribution must be handled in such a way that output is not delivered to someone who might be considered a security risk.

Evaluation of security measures examines:

- the processes involved;
- the data flow;
- the controls;
- the responsibilities for input data;
- the responsibility for storing and maintaining data;
- the ability of processes to accept data that is imprecise, incomplete, or not current;
- the impact of loss or destruction of data;
- the possible advantages that might accrue to an unauthorized user who has illegally obtained data;
- the proposed accident prevention measures;
- the proposed reconstruction routines;
- the expected volumes of stored and maintained data; the storage media for data;
- the expected update frequency for data;
- the number and skills of operators; and
- the number and skills of maintenance programmers.

Benefits of the proposed system. One of the first uses of the development proposal is to perform an economic analysis of the proposed system. This means that the benefits that the proposed system will provide to the user entity and to the customer entity also have to be stated. This criterion measures these benefits. In recording the expected benefits of the proposed system we obtain a list of the user expectations.

The benefits should be listed without assigning values to them; value judgment should be performed later during the economic analysis. By delaying the economic analysis one avoids the problem of juggling figures in order to justify a proposed system that is inappropriate.

The benefit evaluation examines:

- the operational difficulties of the present system;
- the operational difficulties of the proposed system;
- the security subsystem of the present system;
- the security subsystem of the proposed system;

- the number of unsatisfied service requests in the present system;
- the expected volume of service requests; the expected growth in service requests; the estimated changes in personnel;
- the dependency of the user on availability of equipment and external people; and
- the list of proposed benefits, showing not only the benefits but also giving brief descriptions of their importance, without quantifying it.

4.3 The Relationship of the Proposed System to Environment

In this appraisal class the interaction of the proposed system with other systems, either operating or under development, is evaluated. The appraisal criteria for this class are singularity, integrability, use by other groups, and global benefits and priority.

Singularity. This criterion determines whether the proposed system is unique or quite similar to other systems already in use or being planned.

Instead of constructing a new system, it is often more economical to change an existing system to perform the new functions. Often systems can be developed by changing or using commercial software packages.

The evaluation of this criterion must examine:

- the existence of similar systems within the users' institution;
- the existence of a similar system on the software market;
- the estimated effort required to adapt systems that already exist;
- the difficulty of maintaining a system that was not developed in the home installation;
- the satisfaction of user needs with an acquired system; and
- the ability to generalize the proposed system to incorporate more users.

Integration of the proposed system with other systems. This criterion measures whether data can be interchanged among the proposed system and other systems being developed or in operation.

The current trend is toward overall top-down design of integrated systems [5], but this is not always possible owing to the very nature of the organization for which the system will be developed. In such cases it might be necessary to integrate a system bottom-up or "sideways."

Whenever a total plan is not possible the systems should be designed so that they may interchange data at some later moment without too much effort devoted to adaptation. Of course, operations that are required for integrated systems will still be difficult to implement if the operations cross system boundaries.

The evaluation requires knowledge about the following:

- the existence of a data directory/dictionary [11,12];
- the existence of systems sharing some input data;
- the existence of systems with common user subgroups; and
- the estimated effort to achieve data commonality for systems sharing some input or maintained data.

The capability of the system to be used by other users. This criterion attempts to determine the existence of potential user groups that have not yet been recognized. In

some organizations copies of a system may be used by different user groups, for example, in a company with similar branch offices. The needs of each user group may vary, but it is still possible to develop a single system for all user groups. In a centralized development and maintenance facility, costs to develop systems for all user groups can often be reduced. Such multiple-use systems would include operating systems, compilers, software tools, and multiple installation systems such as might occur in branch plants or offices. The logistics of such development and the maintenance of the various versions of the software are quite difficult and must be known in advance.

The evaluation of this criterion requires knowledge regarding:

- the existence of user groups with similar job goals;
- the effort required to adapt or customize the system to each of the user groups; and
- the difficulty of communicating with different user groups.

Global benefits and priorities. This criterion determines the priority of the proposed system with regard to other systems that are under development or are being considered for development.

Most organizations have a limited capacity to develop new systems; budgetary and manpower constraints are usually the limiting factors. Furthermore, in many organizations there are several projects under development at any one time. Since software projects require large amounts of time, money, and manpower, a scheduling problem often arises. The scheduling problem is even more complicated if new development proposals may arrive at unpredictable times.

Evaluation of this criterion examines:

- the benefits of the proposed system for the user;
- the relative importance of the proposed system for the organization;
- the expected benefits for the organization; the impact on other projects;
- the efficiency and effectiveness of the present system; and
- the risks involved in implementation and operation of the proposed system.

4.4 Operational Feasibility of the Proposed System

This appraisal class determines the ability of the user to operate the proposed system effectively. Many design alternatives depend on operational restrictions. Such restrictions should be explicitly stated; otherwise the resulting system will either not be used, or will not adequately do the job for which it was designed. The investment in a software project is too large to allow such implementation failures.

Many awkward psychological problems, such as resistance to change and willful misuse, are often the result of a system that harasses rather than assists a user. This harassing effect is frequently a result of inadequate interface or dialogue design, unavailability of the computer system when required, or unreasonable restrictions on the user entity [13].

This appraisal class contains the following criteria: the operational cost, the computation restrictions, the operational environment, and the operational security.

Operational cost. This criterion measures the cost restrictions that apply to a system once it becomes operational. The cost of operating a system is related to its complexity, its processing volume, and its required response times, as well as to other parameters.

One should not just state an operational cost; rather a budget should be produced for the operation of the system and the budget should be adjusted to fit the operational cost limit. Of course, if the budget must be reduced, the capabilities of the system are usually reduced in a corresponding manner.

The evaluation of this criterion requires knowledge regarding:

- the number and skills of the persons operating and maintaining the system;
- the proposed security subsystem;
- the proposed data acquisition subsystems;
- the proposed data maintenance subsystems;
- the volume of stored and maintained data;
- the processing volume;
- the response-time restrictions;
- the life expectancy of the proposed system;
- the growth expectancy of the proposed system;
- the estimated probability of not achieving the goals desired of the system;
- the estimated equipment cost; the estimated manpower cost;
- the estimated maintenance cost; and
- other costs, such as those for forms, materials, and rent.

Some cost figures in this list depend upon decisions that should not be made at this stage. For example, different choices of equipment make an impact on costs in different ways. However, one should refrain from choosing equipment at this stage. The total impact of equipment costs seems to be small (20%-30% [14]) in present-day systems, and so their total effect on these cost figures should not be too significant.

Computation restrictions. This criterion determines the execution-time constraints and the computational requirements. The structure of both the program and its data may significantly impact both the execution times and storage requirements, even though the machine is not yet defined. Restrictions on response time and privacy and secondary storage requirements can be applied. These restrictions will then direct the implementation strategies. Obviously, once the equipment is defined these estimated values must be recomputed.

The evaluation of this criterion requires information about the following:

- the expected response time from service-request arrival to output distribution;
- the number of functions required to serve a request;
- the estimated response times per function;
- the storage requirements per function;
- the estimated storage requirements for the data base by storage media;
- the complexity of the proposed system;
- the processing volume per service request;
- the volume of service requests per time unit;
- the functions of the proposed system;
- the data flow of the proposed system;
- the data acquisition subsystem; the security subsystem;
- the data maintenance subsystem; and the estimated equipment cost.

The operational environment. This criterion evaluates the constraints that are independent of the equipment chosen for the proposed system. The operational environment has an impact on security and user satisfaction. Operators and maintenance groups may be dissatisfied with the system if it has not been designed appropriately for their use. Thus an overly complex dialogue between the system and an operator may have a stultifying effect on the operational and direct user. Similarly, too simple a dialogue may have a similar effect on the same type of user who has some background in computing [13, 15].

The operational environment also has an impact on data-acquisition methods since the choice of method affects both operation and development costs. The environment must be determined in a preliminary manner when the development proposal is constructed; however, the equipment and procedures to be used should not be determined at that time.

The background and knowledge of the operators and maintenance programmers can severely influence the style of the documentation and amount of training material to be produced. For example, the documentation may be terse and allow a certain amount of creativity, or be verbose and presented in a "cookbook" manner.

Evaluation of this criterion requires knowledge regarding:

- the number and skills of the operators;
- the number and skills of the maintenance programmers;
- the skills of direct users;
- the expected number of replacement personnel to be hired in a given time period;
- the responsibility level of the operative user;
- the contact that operative or direct users will make with the other users;
- the security subsystem;
- the complexity of the proposed system;
- the functions of the proposed system; and
- the data flow.

Operational security. This criterion measures the suitability of the proposed security subsystem. Data and programs are valuable resources. Their disclosure to others might cause losses, lawsuits, and other similar consequences. Data or program errors may also cause losses, and these might even lead to bankruptcy or loss of life. If security measures are not built into a system from the beginning, their inclusion at a later time might be quite costly, since it may be necessary to rewrite a major portion of the system.

- The evaluation of this criterion requires information about these factors;
- the dependence of the functions on complete, precise, and current data;
- the functions of the proposed system; the data flow;
- the importance of the system to the user;
- the required privacy and confidentiality of data manipulated by the proposed system;
- the estimated losses when data or programs deteriorate or are destroyed;
- the estimated losses if data or programs are disclosed to unauthorized persons;
- the existence of measures to prevent catastrophes;

- the existence of internal and external auditing functions;
- recovery procedures; and
- emergency operation procedures, such as off-line execution of the system when the online system is not operating.

4.5 Feasibility of Implementation

This appraisal class evaluates the adequacy of the embryonic development plan for the proposed system. Such a simplified plan is needed to enable a reasonable estimate of development costs. This plan, of course, is still only a first approximation and will undergo many subsequent changes. These modifications may affect development cost, operational cost, and other costs and schedules associated with the proposed system. Such changes imply that the development cycle should be such that these costs could be reevaluated at appropriate points in time [5].

The development proposal should be regarded as a contract, and hence an effective change-control procedure should be applied whenever changes to the proposal are necessary. This change-control procedure should be flexible; otherwise the development proposal may inhibit development or will be disregarded entirely. At the same time, the change-control procedure should not be so flexible that it could allow too frequent or contradictory changes. Since the development proposal does not expose the system in detail, it is expected that changes will not occur too frequently.

This appraisal class contains the following evaluation criteria: technical feasibility, schedule feasibility, manpower feasibility, support feasibility, financial feasibility, development security, and additional clauses.

Technical feasibility. This criterion measures the technical difficulty of implementing the proposed computer system. The evaluation requires knowledge regarding:

- the function of the proposed system; the data flow;
- the security subsystem;
- the expected elapsed time per service request;
- the expected processing volume;
- the expected volume of service request;
- the number of functions per service request;
- the expected response time per function;
- the data-acquisition subsystem;
- the data-maintenance subsystem;
- the volume and storage media of maintained data;
- the estimated complexity of the system;
- the estimated amount of innovation required in the system; and
- the expected professional experience of the people involved in the development.

Certain of the parameters may indicate the existence of real-time constraints. When the time necessary to perform the service is almost equal to the expected elapsed time, any operational difficulty can make it impossible to accomplish the service within the stated time constraints. If such cases exist, emergency measures must be anticipated.

Schedule feasibility. This criterion measures the adequacy of the proposed schedule for the development of the system. The evaluation should consider:

- the phases of the development schedule-names, descriptions, and products;
- the estimated duration of each phase;
- the estimated technical and administrative manpower required for each phase;
- the estimated computational resources for each phase;
- the estimated requirements for other resources, such
- as special equipment, materials, rentals, and documentation;
- the complexity of the proposed system;
- the estimated amount of technological innovation required;
- the possible difficulties that might arise during each phase-that is, the risk factors; and
- the acceptance for each phase.

Manpower feasibility. This criterion measures the adequacy of the estimated manpower requirements. The evaluation requires information regarding:

- the phases of the proposed development plan;
- the functions of the proposed system;
- the number and skills of technical manpower required;
- the number and skills of clerical manpower required;
- the complexity of the proposed system;
- the estimated amount of technological innovation required;
- the expected professional experience of the manpower required;
- the expected professional skill for each phase;
- the development security subsystem; and
- the operational security subsystem.

Support feasibility. This criterion measures the adequacy of the support that would be provided during the development of the system. The evaluation should consider:

- the phases of the proposed development plan; the complexity of the proposed system;
- the estimated amount of technological innovation;
- the expected professional experience of the manpower required;
- the estimated computational-support requirements for each phase;
- the estimated administrative-support requirements for each phase; and
- the other estimated support requirements for each phase.

Financial feasibility. This criterion measures the adequacy of cost proposals and whether cash flow will be adequate during the development of the proposed system. The evaluation requires knowledge regarding:

- the phases of the proposed development plan;
- the estimated cost of each phase;
- the estimated manpower for each phase;
- the estimated support requirements for each phase; the estimated duration of each phase;
- the acceptance criteria for each phase; and

- the payment schedule and its links to the development plan.

Development security. This criterion measures the adequacy of the security surrounding the development effort. The operational security might include restrictions on the development activity. For example, if a disclosure of programs would be a significant risk to the company, implementation security might require the creation of additional tools such as programs.

The evaluation of this criterion should consider:

- the operational security subsystem;
- the internal and external operational audits;
- the development manpower;
- the responsibility level of the development manpower;
- the disclosure risk for programs and test data;
- the expected personnel turnover during development;
- the functions that are critical to security;
- the functions that are critical when modified in a transparent manner; (For example, does a function behave adequately but contain non specified extensions that might benefit others?)
- the existence of recovery procedures for the development;
- the existence of procedures for preventing catastrophe during development; and
- the existence of change procedures.

Additional clauses. This criterion measures the necessity of any additional contractual clause. Such a clause could have a significant impact on a development proposal; for example, the necessity of a maintenance period in the contract, personnel clearance, acceptance criteria, site test requirements, manpower selection, and specific penalty payments if schedules are not observed.

The change control to be used during development of the proposed system should be instituted at this stage. It should be emphasized that such a change control is a necessity since otherwise the development proposal may become just another piece of paper in a filing cabinet.

5 Epilogue

The utility and necessity of the development proposal have been discussed, and an attempt has been made to define its contents and review process. An outline for the contents of this proposal is provided as an Appendix. This outline may be too extensive in some areas, particularly for the development of small systems. In such a case it should be adapted appropriately.

The development proposal as stated here is a revision of another guideline [7]. This earlier development-proposal guideline has been used to define small (less than 1 person-year) application systems. Several minor flaws have been observed and, hopefully, eliminated in this current version.

This version of the development proposal has been applied to another set of small projects. The major drawback noted was "excessive" rigor since the cost to produce the development proposal as stated herein compared to the size of the projects was rela-

tively high. However, agreement was attained in that a clear understanding of the project was gained, and owing to this understanding the following development phases did not run into trouble. It has not yet been possible to obtain a controlled quantitative assessment.

Appendix: Development proposal outline

The body of this paper has described the reasons for a development proposal and its method of construction. This Appendix presents an outline (Table Al) for a sample development proposal, giving titles for each chapter of the report and, in most cases, section and subsection headings.

This Appendix could be used as a guideline for constructing a development proposal, although the various section and subsection headings would have to be adapted to the system being proposed. Hopefully most of the headings will be self-explanatory, but comments are interspersed in the text where appropriate.

The subsection headings are often given in the form of questions. This form is used to emphasize that the individuals constructing the development proposal must seek information which to answer the specific queries adequately.

It should be noted that whenever the pronoun "who" is used in this Appendix a manpower position, rather than a specific individual, is denoted.

Table Al. Outline for Development Proposal

1. Summary

2. Present system

2.1 Tasks

- Tasks performed by the present system
- Goals of these tasks
- Data flow between tasks
- Data required to activate tasks
- Volume of input data per task
- Volume of results per task
- Number of operational steps (maximum/minimum) required per task, including sorting, aggregating, computing, and similar operations
- Nature of the present systems (Is the present system centralized, decentralized, computerized, partially automated or manual?)
- The importance of the results produced by the system.

2.2 Service requests

- What are the service requests?
- Who originates the service requests: a person, an event, or a time dependency?
- How are service requests originated?
- The frequency of occurrence?
- Tasks that must be performed to provide the required services?
- Expected results per service request?

- Expected and real response time from service-request arrival to output distribution?
- What is the number (volume or frequency) of service requests left unsatisfied and the major reasons for lack of service?

2.3 Data

- Who collects and prepares data?
- Who is responsible for the maintenance of data?
- What are the storage media for data maintained by the system?
- What are the access or retrieval mechanisms and their inherent difficulties?
- What is the frequency of occurrence of incorrect, incomplete, or noncurrent data?
- How well does a task function using incomplete, imprecise, or noncurrent data?
- Description of the forms used by the present system

2.4 Controls

- What methods are used to determine whether the input and stored data are correct and complete?
- What methods are used to determine whether the results are complete?
- What methods are used to determine the currency of the data? What are the correction and recovery procedures?

2.5 Security

- What are the catastrophe prevention measures?
- How confidential are the data manipulated by the system?
- What is the degree of privacy for maintained data?
- Who has access to the data under different conditions?
- What are the existing recovery mechanisms?
- What are the possible consequences of unauthorized access?
- What are the possible consequences of accidents (fire, equipment failure)?
- What are the possible consequences of unauthorized modification to the data?

2.6 User entity

- Operative user
- Skills and training
- Training time
- Rate of attrition
- Organization
- Categories of operative users
- Direct user
- Skills and training as related to the present system Opportunity to train
- Rate of attrition
- Examples of categories and uses
- Indirect user
- Access to the system

- Examples of uses

2.7 Importance of the present system

- How important is the present system in achieving the objectives of the user entity?
- What losses occur as a consequence of inadequacies in the present system?
- How important is the present system in achieving the objectives of the customer entity?

3. Proposed system

3.1 Functions

- Functions (manual or computerized) to be performed by the proposed system
- Objectives of these functions
- Data flow among functions
- Necessary data to activate functions
- Estimated volume of input data per function
- Estimated volume of results per function
- Estimated processing volume per function
- Importance of the results to be produced by the proposed system
- Nature of the proposed system (centralized, decentralized, on-line, batch, or distributed)

3.2 Service requests

- What are the proposed service requests?
- Who originates these service requests?
- How are service requests originated?
- Frequency of occurrence
- When are service requests originated?
- Expected results per service request
- Expected response time per service request
- Controls applying to service requests

3.3 Data

- Who will gather and prepare data?
- Who will be responsible for maintenance of the data? What are the proposed storage media?
- What is the expected data volume at installation and what is the estimated capacity?
- How well will functions perform with data that are incomplete, imprecise, or not current?

3.4 Controls

- What methods are used to determine whether the input and stored data are correct and complete?

- What methods are used to determine whether the results are correct and complete?
- What methods are used to determine the currency of the data?
- What are the proposed correction and recovery procedures?

3.5 Security

- What are the proposed measures for prevention of a catastrophe or accident?
- How confidential are the data to be manipulated by the proposed system?
- What is the degree of privacy for the maintained data?
- Who can read or modify the data?
- What are the planned recovery mechanisms?
- What are the possible consequences of losses?
- What are the expected reasons (benefits, vengeance) of unauthorized access?
- Emergency operation procedures.

3.6 User entity

- Operative user
- Expected skills and training
- Expected time to train
- Expected rate of attrition
- Proposed organization
- Examples of categories
- Direct user
- Expected skills and training in the use of the proposed system
- Opportunity to train
- Expected rate of attrition
- Example of categories
- Indirect user
- Proposed access to the system
- Examples of uses

3.7 Expectations

- Date of initial operation
- Minimal cost to develop
- Minimal cost to operate over a period of time
- Definite equipment requirements
- Expected data growth
- Expected growth of functions
- What are the dependencies to be tolerated (portability, languages, operating systems)?
- Benefits of the proposed system-statement and justification.

4. Relation of proposed system to environment

4.1 Uniqueness of proposed system

- A brief overview of existing similar systems, and/or proposed alternative systems, within the installation and on the market
- The effort required to adapt existing systems to fulfill user needs
- The expected difficulty of maintaining an adapted system The expected user satisfaction using an adapted system The potential additional users of the proposed system

4.2 Integration of proposed system

- Does a data directory/dictionary system exist?
- A brief overview of other systems that might or will share data A brief overview of other systems that might share direct or indirect users
- The expected effort to achieve data commonality for systems sharing data

4.3 User expansion

- The existence of user groups with similar job goals
- The effort required to encompass these additional user groups The difficulty of communicating with user groups

4.4 Global benefits and priorities

- The relative importance of the proposed system for the organization
- The expected benefits for the organization-description and justification
- The impact of the development of the proposed system on other ongoing development efforts
- The risks involved in implementation and operation of the proposed system

5. Feasibility of implementation-proposed system

5.1 Implementation risk

- An estimate of the amount of innovation required in the system
- The existence (or lack) of development tools and aids
- The professional experience of the available manpower, with specific reference to the proposed system

5.2 Development process

- The proposed development phases
- The proposed results for each phase
- The proposed main milestones
- The proposed acceptance criteria for each result and each phase
- The proposed method to control modifications

5.3 Preliminary schedules

- Schedule for phases
- Manpower schedule-types of manpower should be shown
- Computational resources schedule
- Support schedule-types of support should be shown Budget
- Product schedule

5.4 Development security

- Catastrophe and accident prevention and recovery during development
- Disclosure risk for programs and test data
- Functions that must be secure
- Expected manpower turnover during development
- External and internal audits during development
- Expected degree of responsibility of manpower

5.5 Additional clauses

- Additional clauses, should be listed here and a brief justification of their necessity and importance be given.

REFERENCES

1. E. R. Mangold and R. D. Williams, "Software Visibility and Management", in *Proceedings of the TRW Symposium on Reliable, Cost-Effective, Secure Software* (B. Boehm and E. Gaynor, eds.), TRW-SS-74-14, TRW, Redondo Beach, California, March 1974, pp. 201-265.
2. P. W. Metzger, *Managing a Programming Project*, Prentice-Hall, Englewood Cliffs, N.J., 1973.
3. W. W. Royce and D. J. Allen, "Software Requirements Analysis, Sizing and Costing", in *Proceedings of the TRW Symposium on Reliable, Cost-Effective, Secure Software* (B. Boehm and E. Gaynor, eds.), TRW-SS74-14, TRW, Redondo Beach, California, March 1974, pp. 1.1-1.77.
4. G. F. Hice, W. S. Turner, and L. F. Cashwell, *System Development Methodology*, North Holland, Amsterdam, 1974.
5. T. R. Gildersleeve, *Data Processing Project Management*, Van Nostrand Reinhold, New York, 1974.
6. A. von Staa, *Desenvolvimento Estruturado de Sistemas Automatizados* (Structured Development of Automated Systems), Monografias em Ciência da Computação/79, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil, 1979.
7. A. von Staa, *Metodologias para o Desenvolvimento de Sistemas de Informação: Proposta de Desenvolvimento*, Departamento de Informática, Pontifícia Universidade Católica, Rio de Janeiro, Internal report, 1977.
8. International Business Machines, Inc., *Business Systems Planning: Information Systems Planning Guide*, IBM Corp., New York, August, 1975.
9. D. Teichroew and H. E. Hershey, "PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems", *IEEE Transactions on Software Engineering* SE-3 (1), 41-49 (1977).
10. *IEEE Transactions on Software Engineering*. SE-3 (1), various articles (1977).
11. C. J. Bontempo, "Data Resource Management", *Data Management 11*, 33-37 (February 1973).

12. B. K. Plagman and G. P. Altshuler, "A Data Directory Dictionary System Within the Context of an Integrated Corporate Data Base", *AFIPS Fall Joint Computer Conference*, 1972, Vol. 21, Part II, pp. 1133-1140.

13. J. Martin, *Design of Man-Computer Dialogues*, Prentice-Hall, Englewood Cliffs, N.J., 1973.

14. B. Boehm, "Software Engineering", *IEEE Transactions on Computers* C-25 (12), 1126-1241 (1976).

15. S. Woolridge, *Computer Input Design*, Petrocelli/Charter, New York, 1974.