'HKIS 569

Printing Requisition/Graphic Services

Dept. No. 51892

Title or Description
CS 78-31 THE DEVELOPMENT PROPOSAL: THE FIRST STEP IN SOFTWARE SYSTEM CONSTRUCTION

Date: 19 September/79
Date Required: ASAP
Account: 126-4100-01

Signature
Signing Authority

Department: Computer Science
Room: 5179
Phone: 3143
Delivery: ☒ Mail □ Pick-up □ Via Stores □ Other

1. Please complete unshaded areas on form as applicable. (4 part no carbon required).
2. Distribute copies as follows: White, Canary and Pink—Printing, Arts Library or applicable Copy Centre Goldenrod—**Retain.**
3. On completion of order, pink copy will be returned with printed material. Canary copy will be costed and returned to requisitioner, **Retain as a record of your charges.**
4. Please direct enquiries, quoting requisition number, to Printing/Graphic Services, Extension 3451.

Reproduction Requirements: ☒ Offset □ Signs/Repro's □ Xerox
Number of Pages: 74
Number of Copies: 55

Type of Paper Stock: ☒ Bond □ Book □ Cover □ Bristol □ Supplied Covers & Backs Supp

Paper Size: ☒ 8½ x 11 □ 8½ x 14 □ 11 x 17

Paper Colour: ☒ White □ Other
Ink: □ Black

Printing: □ 1 Side ☒ 2 Sides
Numbering to

Binding/Finishing Operations: ☒ Collating □ Corner Stitching □ 3 Ring □ Tape □ Plastic Ring □ Perforating

Folding Finished Size 3 STAPLES
Cutting Finished Size

Special Instructions
Please have the enclosed covers attached to the report, stapled on the top, middle and bottom. Thnks.

Cost: Time/Materials	Fun.	Prod.Un.	Prod. Cl.	Opr. No.	Mins.	Total
Signs/Repro's	1					
Camera	2					
Correcting & Masking Negatives	3					
Platemaking	4					
Printing	5			3	60	
Bindery	6			21	-30	
53				22	-30	
Sub. Total Time						
Sub. Total Materials						
Prov. Tax						
Total						

Film Qty	Size	Plates Qty	Size & Type
Paper Qty	Size	Plastic Rings Qty	Size
Outside Services			

Graphic Services May 68 482-2

THE DEVELOPMENT PROPOSAL: THE FIRST STEP IN SOFTWARE SYSTEM CONSTRUCTION

Arndt von Staa[1]
Departamento de Informatica
Pontificia Iniversidade Catolica
Rio de Janeiro, Brasil

Donald D. Cowan[2]
Computer Science Department
University of Waterloo
Waterloo, Ontario, Canada

Research Report CS-78-31

Revised June 1979

[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.

THE DEVELOPMENT PROPOSAL: THE FIRST STEP IN SOFTWARE SYSTEM CONSTRUCTION

Arndt von Staa[1]

Departamento de Informatica, Pontificia Iniversidade Catolica,
Rio de Janeiro, Brasil.

Donald D. Cowan[2]

Computer Science Department, University of Waterloo, Waterloo, Ontario,
Canada.

Typed by: Alana V. Bachmann on the IBM MC 82.

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 called 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.

[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.

June 8, 1979.

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 [MAW 74; MET 73] 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-brush definition [ROA 74] since it is the first step in the refinement process.

We call this first document describing the software system "The Development Proposal" and we 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:

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

are important to the system and should be answered early in its development. If 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 will not enter into more detail, referring the interested reader to [MET 73, HTC 74, GIL 74, STA 79].

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. Section 3 identifies the groups of people or entities involved in developing and using a computerized system. In Section 4 an outline of the method for constructing and reviewing the development proposal is presented. Section 5 describes the review criteria used to evaluate the development proposal. The final section reviews the entire concept of a development proposal. An outline of the development proposal by chapter, section and sub-section is presented in the Appendix. This Table of Contents 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 [MAW 74, ROA 74, MET 73, HTC 74]. 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 [ROA 74].

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 less 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 which 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 the system, to be developed. If the proposal does not exist, we might end up with a patchwork which 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, first, it would introduce excessive detail (the analysis does not contribute sufficiently to the understanding of the proposal) and, second, 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; hence it determines the level of knowledge at the time the proposal was prepared. The review is not merely an acceptance criterion, but an explanation of the important portions of the proposal, and 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 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 and its main purpose is to serve as an agreed 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 one [STA 77]. 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 [IBM 75] 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, nature and geographical dispersion of the company. The business system planning method is very formalised 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 [TEH 77] proposes a documentation system and a formal mechanised 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 [ROA 74, HTC 74, GIL 74, MET 73] mention the need for a development proposal and give some details about its contents. In the January 1977 issue of IEEE Transactions on Software Engineering SE-3(1), several articles concerning Requiremant Analysis are presented. Again the main concern seems 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 respond adequately to each item in the guidelines, and so production of a final version is likely to be an iterative process.

Once the first version of a development proposal has been produced, it is reviewed by the co-ordinator (system designers, system architects) and by the users or their representatives, and a review report is produced. Using the review report and the development proposal, it is the co-ordinator'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 co-ordinator. This process continues until no more changes can be made, and both users and the co-ordinator 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 3.1 we show steps in the construction and review process for a development proposal.

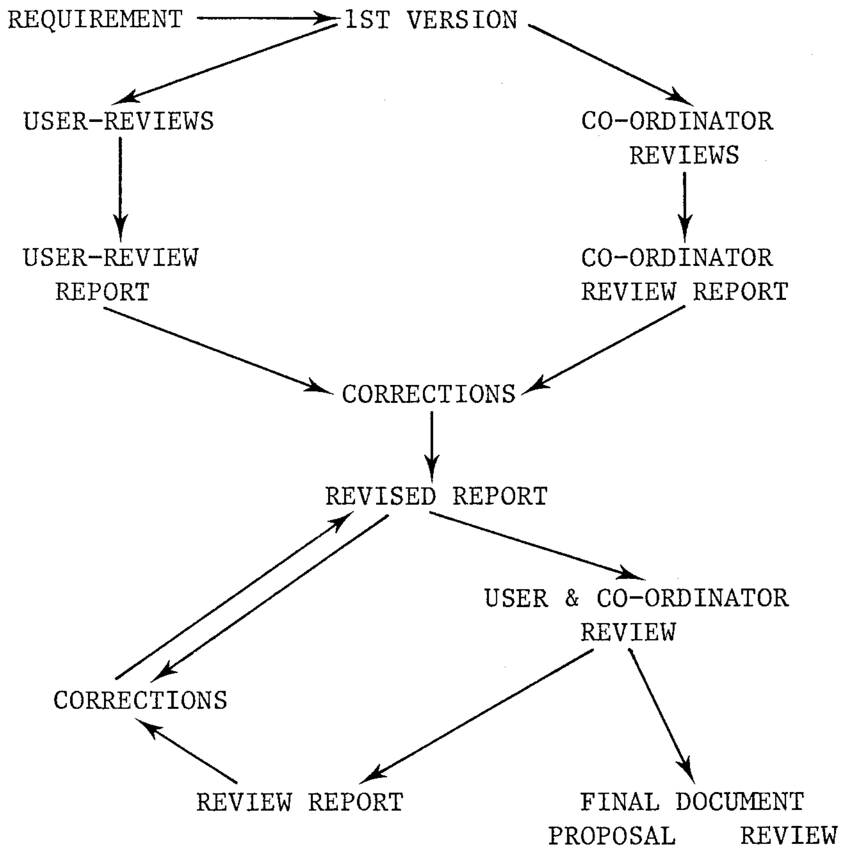

Figure 3.1 Construction and Review Process

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:

(i) The user and co-ordinator cannot agree on a final proposed system because the users are unable to specify the requirements of the system.

(ii) The corrections to the development proposal negate previous corrections as no adequate record-keeping mechanism for changes has been instituted.

(iii) The use of excessive detail prevents completion of the proposal since the users and co-ordinator do not understand its purpose.

(iv) The system is developed independent of its final use; this is often the situation with systems developed as status symbols.

(v) Neither the users nor the co-ordinator 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 and a typical Table of Contents is in the Appendix. The group preparing the development proposal amplifies each section and sub-section of the Table of Contents.

The review or evaluation method is presented in this section. As an evaluation procedure, the review must answer questions about the original description of the system to determine the description is complete. The answers to these questions often pinpoint deficiences 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 which 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 which attempt 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:

(i) The current system.

(ii) The proposed system.

(iii) The relation of the proposed system to the environment.

(iv) The operational feasibility of the proposed system, and,

(v) The feasibility of developing the proposed system.

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

Several of the appraisal criteria overlap one another, and hence contain some redundancy. Redundancy is a desirable property since it provides a means for checking results, and hence provides 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 sub-divided into: (a) description, (b) current organization, (c) necessity, (d) complexity, (e) security.

4.1.1 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: (a) the tasks performed by the current system - both manual and computerized; (b) the objective of these tasks; (c) the data flow between tasks; (d) the service requests which may occur, and which tasks are triggered by these requests; (e) the origin of the service requests; (f) the periodicity, or volume per time unit, of service requests; (g) the results required for each type of service request; (h) the external inputs per task; (i) the data maintained by the current system; (j) the storage media such as ledgers, file cabinets, tapes and disks; (k) the expected and real elapsed time from service-request reception to output distribution, and (l) the number of service requests which are not satisfied, and main reasons for this lack of satisfaction.

4.1.2 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 an inadequate organization. Installing a computerized system in such a case will not alleviate these organizational difficulties and in fact it 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: (a) the tasks performed by the present system; (b) the data flow between tasks; (c) the controls which are imposed on the operation of the present system; (d) the operational difficulties of the present system; (e) the volume of stored data; (f) the storage media used for the data; (g) any access and retrieval difficulties; (h) the number of operational users involved in the operation of a task; (i) the number and complexity of processing steps for each service request; (j) 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 man-hours; (k) the number of service requests which have not been properly processed and the reasons; (l) the real and expected elapsed time between service requests arriving, and output distribution; (m) the number, size (number of data items) and complexity of forms used by the present system; (n) the operating cost of the present system; (o) the volume of service requests, and (p) examples of each class of user, emphasizing trianing and skills.

4.1.3 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 which is really operating in a satisfactory manner.

4.1.4 Current System Complexity

The evaluation requires information about: (a) the tasks performed by the current system; (b) the data flow in the current system; (c) the effect of data which is imprecise, incomplete and not current on the tasks of the system; (d) 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 inter-dependency of the stored data - should be examined; (e) the number of forms required; (f) the volume of services required; (g) the expected and real elapsed time from service-request arrival to to output distribution; (h) the number of steps required to produce desired outputs per service request, and (i) the operative user - the training and skills required.

4.1.5 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: (a) the ability of a task to function with data which is incomplete, imprecise or not current; (b) the degree of confidentiality of the data - what happens if some data is known to unauthorized personnel?; (c) the direct and indirect users' view of the importance of the present system; (d) the volume of stored data; (e) the storage media used for maintaining data; (f) the existing catastrophe protection measures; (g) the existing access authorization measures; (h) the difficulty of reconstructing destroyed data; (i) the frequency with which data is incomplete, imprecise or not current, and (j) 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 not realistic.

The appraisal criteria for the proposed system may be sub-divided into:

(a) Definition of the proposed system.

(b) Definite restrictions on the proposed system.

(c) The life expectancy of the proposed system.

(d) The complexity of the proposed system.

(e) The complexity of the solution to the problem relative to the task to be performed.

(f) Security of the proposed system, and

(g) Benefits of the proposed system.

4.2.1 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 which shows how the proposed system will operate. Obviously, this preliminary concept is likely to be quite different from the system design which 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 processess which will have to be designed [ROA 74]. 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: (a) a statement of the automatic and manual processes contained within the proposed system; (b) the description of these processes; (c) the data flow between the processes; (d) the necessary data to activate each individual process; (e) the output for each process; (f) the data maintained by the processes, specifically the type of data but not the implementation details; (g) the service requests and their descriptions; (h) the origin of service requests; (i) the frequency, or volume per time unit, of each service request; (j) the expected elapsed time or response time from service-request arrival to output distribution; (k) the responsibility for input data; (l) the responsibility for storing and maintaining data; (m) the need for the outputs; (n) the security requirements for input data; (o) the security requirements for output data; (p) 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 online files, dismountable files, filing cabinets and ledgers, and (q) the expected volumes of inputs and outputs, data bases and service requests.

4.2.2 Definite Restrictions

This criterion determines the validity of any pre-imposed design and implementation alternatives.

Pre-imposed 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 pre-imposed 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: (a) restriction description, (b) justification for the restriction.

Typical restrictive items are : (i) hardware; (ii) portability; (iii) operating systems; (iv) software support systems; (v) languages; (vi) input/output equipment; (vii) forms; (viii) labels, density, and layout for each file; (ix) development procedures; (x) documentation procedures; (xi) programming style; (xii) system types available - such as online systems, centralized systems, and networks; (xiii) quality and maintenance requirements, and (xiv) 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.

4.2.3 Proposed Organization

This criterion measures the impact a proposed computer system will have on the user organization, and 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 user must be made aware initially of his/her responsibility in both the areas of implementation and operation of the system.

The evaluation of the proposed organization should consider: (a) the processes provided by the proposed system; (b) the data flow; (c) the groups responsible for producing input data; (d) the group responsible for storing and maintaining data; (e) the outputs of the system and the destination of these outputs; (f) the expected use to be made of these outputs; (g) the storage media for the data; (h) the mechanisms to gather and transcribe data into machine-readable form; (i) the number and skills of all operations personnel; (j) the number and skills of all maintenance personnel; (k) the controls of the system including security and quality control; (l) the expected elapsed time between input and receipt of output - the so-called response time; (m) any equipment restrictions; (n) any software restrictions, and (o) programming restrictions.

4.2.4 Life Expectancy

This criterion measures the expected life-time 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 should be enhanced, in advance of a crisis. Furthermore, it is possible to enforce expansion criteria for the beginning of a system and thus hopefully reduce costs of unexpected expansion.

An evaluation of life expectancy should consider: (a) the expansion policy of the user entity; (b) the integration and centralization policies of the user entity; (c) the expected growth of the data base; (d) the expected growth of service requests; (e) the expected growth of future additions to the system; (f) the current volume of the data base, and (g) the current volume of the service requests.

4.2.5 Complexity of the Proposed System

This criterion measures the operational difficulty which 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, while others will be caused by the naivety of the system design. Stating the operational difficulties acts as a design control and is also an indicator of the necessity of a more careful analysis and system design to be performed during later phases.

An evaluation of complexity should consider: (a) the processes involved; (b) the data flow; (c) the origin and responsibilities for data; (d) the responsibility for storing and maintaining data; (e) the data-gathering and data-transcription procedures; (f) the controls; (g) the output distribution; (h) the volume of service requests; (i) the expected maximum elapsed time between service-request arrival and output distribution; (j) the estimated volume of processing per service request; (k) the number of steps required to handle a service request; (l) the ability of the system to handle data which is imprecise, incomplete, and not current; (m) the storage media for data; (n) security against accidents; (o) security against corruption such as the misuse or wilful destruction of data or programs, and (p) general quality requirements .

4.2.6 Simplicity of Solution

This criterion measures whether certain functions, outputs and service requests are really necessary.

The main purpose of this criterion is to verify whether the proposed 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 Sub-section 4.1.4. One should also include: (a) the expected use of outputs, and (b) expected users of system output.

4.2.7 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 which 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: (a) the processes involved; (b) the data flow; (c) the controls; (d) the responsibilities for input data; (e) the responsibility for storing and maintaining data; (f) the ability of processes to accept data which is imprecise, incomplete or not current; (g) the impact of loss or destruction of data; (h) the possible advantages which might accrue to an unauthorized user who has illegally obtained data; (i) the proposed accident prevention measures; (j) the proposed reconstruction routines; (k) the expected volumes of stored and maintained data; (1) the storage media for data; (m) the expected update frequency for data; (n) the number and skills of operators, and (o) the number and skills of maintenance programmers.

4.2.8 Benefits of the Proposed System

This criterion measures the benefits that the proposed system will provide to the user entity and to the customer entity.

One of the first uses of the development proposal is to perform an economic analysis of the proposed system. This means that the benefits also have to be stated. In recording the expected benefits of the proposed system we obtain a list of the user expectations.

The benefits should be listed without assigning a value to them. The value judgement 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 which is inappropriate.

The benefit evaluation examines: (a) the operational difficulties of the present system; (b) the operational difficulties of the proposed system; (c) the security sub-system of the present system; (d) the security sub-system of the proposed system; (e) the number of unsatisfied service requests in the present system; (f) the expected volume of service requests;

(g) the expected growth in service requests; (h) the estimated changes in personnel; (i) the dependency of the user on availability of equipment and external people, and (j) the list of proposed benefits - this list should show not only the benefit but should also give a brief description of the importance of such a benefit, 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: (a) singularity, (b) integrability, (c) use by other groups, and (d) global benefits and priority.

4.3.1 Singularity

This criterion determines whether the proposed system is unique or whether it is 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: (a) the existence of similar systems within the users' institution; (b) the existence of a similar system on the software market; (c) the estimated effort required to adapt systems which already exist; (d) the difficulty of maintaining a system which was not developed in the home installation; (e) the satisfaction of user needs with an acquired system, and (f) the ability to generalize the proposed system to incorporate more users.

4.3.2 Integration of the Proposed System with Other Systems

This criterion measures whether data can be interchanged between the proposed system and other systems being developed, or in operation.

The current trend is toward overall top-down design of integrated systems [IBM 75], 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 from the 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 which are required for integrated systems will still be difficult to implement if the operations cross system boundaries.

The evaluation requires knowledge about: (a) the existence of a data directory/dictionary [BON 73, PLA 72]; (b) the existence of systems sharing some input data; (c) the existence of systems with common user sub-groups, and (d) the estimated effort to achieve data-commonality for systems sharing some input ot maintained data.

4.3.3 The Ability of the System to be Used by Other Users

This criterion attempts to determine the existence of potential user groups which have not yet been recognized.

In some organizations copies of a system may be used by different user groups, such a situation would occur 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 of: (a) the existence of user groups with similar job goals; (b) the effort required to adapt or customize the system to each of the user groups, and (c) the difficulty of communicating with different user groups.

4.3.4 Global Benefits and Priorities

This criterion determines the priority of the proposed system with regard to other systems which are under development or which are being considered for development.

Most organizations have a limited capacity to develop new systems; budgetary and man-power 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 man-power, 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: (a) the benefits of the proposed system for the user; (b) the relative importance of the proposed system for the organization (c) the expected benefits for the organization; (d) the impact on other projects; (e) the efficiency and effectiveness of the present system, and (f) 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 wilful misuse are often the result of a system which 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 [MAR 73].

This appraisal class contains the following criteria: (a) the operational cost; (b) the computation restrictions; (c) the operational environment, and (d) the operational security.

4.4.1 Operational Cost

This criterion measures the cost restrictions which 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 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 the criterion requires a knowledge of: (a) the number and skills of the persons operating and maintaining the system; (b) the proposed security sub-system; (c) the proposed data acquisition sub-systems; (d) the proposed data maintenance sub-systems; (e) the volume of stored and maintained data; (f) the processing volume; (g) the response-time restrictions; (h) the life expectancy of the proposed system; (i) the growth expectancy of the proposed system; (j) the estimated probability of not achieving the goals desired of the system; (k) the estimated equipment cost; (l) the estimated man-power cost; (m) the estimated maintenance cost, and (n) other costs - such as those for forms, materials and rent.

Some cost figures in this list depend upon decisions which 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 to 30% [BOE 76]) in present-day systems, and so their total effect on these cost figures should not be too significant.

4.4.2 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: (a) the expected response time from service-request arrival to output distribution; (b) the number of functions required to serve a request; (c) the estimated response times per function; (d) the storage requirements per function; (e) the estimated storage requirements for the data base by storage media; (f) the complexity of the proposed system; (g) the processing volume per service request; (h) the volume of service requests per time unit; (i) the functions of the proposed system; (j) the data flow of the proposed system; (k) the data acquisition sub-system; (1) the security sub-system; (m) the data maintenance sub-system, and (n) the estimated equipment cost.

4.4.3 The Operational Environment

This criterion evaluates the constraints which 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 [MAR 73, WOO 74].

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 can be terse and allow a certain amount of creativity, or it may be verbose and presented in a "cook-book" manner.

Evaluation of this criterion requires a knowledge of : (a) the number and skills of the operators; (b) the number and skills of the maintenance programmers; (c) the skills of direct users; (d) the expected number of replacement personnel to be hired in a given time period; (e) the responsibility level of the operative user; (f) the contact that operative or direct users will make with the other users; (g) the security sub-system; (h) the complexity of the proposed system; (i) the functions of the proposed system, and (j) the data flow.

4.4.4 Operational Security

This criterion measures the suitability of the proposed security sub-system.

Data and programs are valuable resources. Their disclosure to others might cause losses, law suits 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: (a) the dependence of the functions on complete, precise and current data; (b) the functions of the proposed system; (c) the data flow; (d) the importance of the system to the user; (e) the required privacy and confidentiality of data manipulated by the proposed system; (f) the estimated losses when data or programs deteriorate or are destroyed; (g) the estimated losses if data or programs are disclosed to unauthorized persons; (h) the existence of measures to prevent catastrophes; (i) the existence of internal and external auditing functions; (l) recovery procedures, and (m) emergency operation procedures - such as offline 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 re-evaluated at appropriate points in time [GIL 74].

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: (a) technical feasibility; (b) schedule feasibility; (c) man-power feasibility; (d) support feasibility; (e) financial feasibility; (f) development security, and (g) additional clauses.

4.5.1 Technical Feasibility

This criterion measures the technical difficulty of implementing the proposed computer system.

The evaluation requires knowledge about: (a) the function of the proposed system; (b) the data flow; (c) the security sub-system; (d) the expected elapsed time per service request; (e) the expected processing volume; (f) the expected volume of service request; (g) the number of functions per service request; (h) expected response time per function; (i) the data-acquisition sub-system; (j) the data-maintenance sub-system (k) the volume and storage media of maintained data; (l) the estimated complexity of the system; (m) the estimated amount of innovation required in the system, and (n) 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.

4.5.2 Schedule Feasibility

This criterion measures the adequacy of the proposed schedule for the development of the system.

The evaluation should consider: (a) the phases of the development schedule - names, descriptions and products; (b) the estimated duration of each phase; (c) the estimated technical and administrative man-power required for each phase; (d) the estimated computational resources for each phase; (e) the estimated requirements for other resources such as special equipment, materials, rentals, documentation; (f) the complexity of the proposed system; (g) the estimated amount of technological innovation required; (h) the possible difficulties which might arise during each phase - that is, the risk factors, and (i) the acceptance for each phase.

4.5.3 Man-power Feasibility

This criterion measures the adequacy of the estimated man-power requirements.

The evaluation requires information about: (a) the phases of the proposed development plan; (b) the functions of the proposed system; (c) the number and skills of technical man-power required; (d) the number and skills of clerical man-power required; (e) the complexity of the proposed system; (f) the estimated amount of technological innovation required; (g) the expected professional experience of the man-power required; (h) the expected professional skill for each phase; (i) the development security sub-system, and (j) the operational security sub-system.

4.5.4 Support Feasibility

This criterion measures the adequacy of the support which would be provided during the development of the system.

The evaluation should consider: (a) the phases of the proposed development plan, (b) the complexity of the proposed system; (c) the estimated amount of technological innovation; (d) the expected professional experience of the man-power required; (e) the estimated computational-support requirements for each phase; (f) the estimated administrative-support requirements for each phase, and (g) the other estimated support requirements for each phase.

4.5.5 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 about: (a) the phases of the proposed development plan; (b) the estimated cost of each phase; (c) the estimated man-power for each phase; (d) the estimated support requirements for each phase; (e) the estimated duration of each phase; (f) the acceptance criteria for each phase, and (g) the payment schedule and its links to the development plan.

4.5.6 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: (a) the operational security sub-system; (b) the internal and external operational audits; (c) the development man-power; (d) the responsibility level of the development man-power; (e) the disclosure risk for programs and test data; (f) the expected personnel turnover during development; (g) the functions which are critical to security; (h) the functions which are critical when modified in a transparent manner, for example, does a function behave adequately but contains non-specified extensions which might benefit others?; (i) the existence of recovery procedures for the development; (j) the existence of procedures for preventing catastrophe during development, and (k) the existence of change procedures.

4.5.7 Additional Clauses

This criterion measures the necessity of any additional contractual clause.

Additional contractual clauses 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, man-power selection, and specific 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. A guideline for the contents of this proposal is outlined in the Appendix. This guideline 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 review of another guideline [STA 77]. This former development-proposal guideline has been used to define small (less than a man-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" rigour since the cost to produce the development proposal as stated herein compared to the size of the projects was relatively 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 quantitive assessment.

APPENDIX

The body of this paper has described the reasons for a development proposal and its method of construction. This Appendix presents a Table of Contents for a sample development proposal. The Table of Contents gives titles for each chapter of the report and in most cases shows section and sub-section headings.

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

The sub-section 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 adequately answers the specific queries.

It should be noted that whenever the pronoun "who" is used in this Appendix it denotes a man-power position rather than a specific individual.

APPENDIX:

TABLE OF CONTENTS

2.1 Present System - Tasks cont'd

system centralized, decentralized, computerized, partially automated or manual?

i - The importance of the results produced by the system.

2.2 Present System - Service Requests

a - What are the service requests?

b - Who originates the service requests, a person, an event or a time dependency?

c - How are service requests originated?

d - The frequency of occurrence.

e - Tasks which must be performed to provide the required services.

e - Expected results per service request.

g - Expected and real response time from service-request arrival to output distribution.

h - What is the number (volume or frequency) of service requests left unsatisfied and the major reasons for lack of service?

2.3 Present System - Data

a - Who collects and prepares data?

b - Who is responsible for the maintenance of data?

c - What are the storage media for data maintained by the system?

d - What are the access or retrieval mechanisms and their inherent difficulties?

e - What is the frequency of occurrence of incorrect, incomplete or non-current data?

f - How well does a task function using incomplete, imprecise or non-current data?

g - Description of the forms used by the present system.

2.4 Present System - Controls

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

b - What methods are used to determine whether the results are complete?

2.4 Present Systems - Controls cont'd

c - What methods are used to determine the currency of the data?

d - What are the correction and recovery procedures?

2.5 Present System - Security

a - What are the measures for prevention of a catastrophe?

b - How confidential is the data manipulated by the system?

c - What is the degree of privacy for maintained data?

d - Who has access to the data under different conditions?

e - What are the existing recovery mechanisms?

f - What are the possible consequences (losses, fire and other problems) of unauthorized access?

g - What are the possible consequences of unauthorized modification to the data?

2.6 <u>User Entity</u>

a - <u>Operative User</u>

i. - skills and training.

ii. - training time.

iii. - rate of attrition.

iv. - organization.

v. - categories of operative users.

b - <u>Direct User</u>

i. - skills and training as related to the present system.

ii. - opportunity to train.

iii. - rate of attrition.

iv. - examples of categories and uses.

c - <u>Indirect User</u>

i. - access to the system.

ii. - examples of uses.

2.7 Importance of the Present System

a - How important is the present system in achieving the objectives of the user entity?

b - What losses occur as a consequence of inadequacies in the present system?

c - How impportant is the present system in achieving the objectives of the customer entity?

3. Proposed System

3.1 Proposed System - Functions

a - Functions (manual or computerized) to be performed by the proposed system.

b - Objectives of these functions.

c - Data flow between functions.

d - Necessary data to activate functions.

e - Estimated volume of input data per function.

f - Estimated volume of results per function.

g - Estimated processing volume per function.

3.1 Proposed System - Functions cont'd

h - Importance of the results to be produced by the proposed system.

i - Nature of the proposed system (centralized, decentralized, online, batch, or distributed).

3.2 Proposed System - Service Requests

a - What are the proposed service requests?

b - Who originates these service requests?

c - How are service requests originated?

d - Frequency of occurrence.

e - When are service requests originated?

f - Expected results per service request.

g - Expected response time per service request.

h - Controls applying to service requests.

3.3 Proposed System - Data

a - Who will gather and prepare data?

b - Who will be responsible for maintenance of the data?

3.3 Proposed System - Data cont'd

c - What are the proposed storage media?

d - What is the expected data volume at installation and what is the estimated capacity?

e - How well will functions perform with data which is incomplete, imprecise or not current?

3.4 Proposed System - Controls

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

b - What methods are used to determine whether the results are correct and complete?

c - What methods are used to determine the currency of the data?

d - What are the proposed correction and recovery procedures?

3.5 Proposed System - Security

a - What are the proposed measures for prevention of a catastrophe or accident?

3.5 Proposed System - Security cont'd

b - How confidential is the data to be manipulated by the proposed system?

c - What is the degree of privacy for the maintained data?

d - Who can read or modify the data?

e - What are the planned recovery mechanisms?

f - What are the possible consequences of losses?

g - What are the expected reasons (benefits, or vengeance) of unauthorized access?

h - Emergency operation procedures.

3.6 Proposed System - User Entity

a - Operative User

i. - expected skills and training.

ii. - expected time to train.

iii. - expected rate of attrition.

iv. - proposed organization.

v. - examples of categories.

3.6 Proposed System - User Entity cont'd

b -Direct User

i. - expected skills and training in the use of the proposed system.

ii. - opportunity to train.

iii. - expected rate of attrition.

iv. - example of categories.

c - Indirect User

i. - proposed access to the system.

ii. - examples of uses.

3.7 Expectations

a - Date of initial operation.

b - Minimal cost to develop.

c - Minimal cost to operate over a period of time.

d - Definite equipment requirements.

e - Expected data growth.

f - Expected growth of functions.

3.7 Expectations cont'd

h - What are the dependencies to be tolerated (portability, languages, operating systems)?

i - Benefits of the proposed system - statement and justification.

4. Relation of Proposed System to Environment

4.1 Uniqueness of Proposed System

a - A brief overview of existing similar systems, and/or proposed alternative systems, within the installation and on the market.

b - The effort required to adapt existing systems to fulfill user needs.

c - The expected difficulty of maintaining an adapted system.

d - The expected user satisfaction using an adapted system.

e - The potential additional users of the proposed system.

4.2 Integration of Proposed System

a - Does a data directory/dictionary system exist?

b - A brief overview of other systems which might or will share data.

c - A brief overview of other systems which might share direct or indirect users.

d - The expected effort to achieve data commonality for systems sharing data.

4.3 User Expansion

a - The existence of user groups with similar job goals.

b - The effort required to encompass these additional user groups.

c - The difficulty of communicating with user groups.

4.4 Global Benefits and Priorities

a - The relative importance of the proposed system for the organization.

b - The expected benefits for the organization - description and justification.

4.4 Global Benefits and Priorities cont'd

c - The impact of the development if the proposed system on other ongoing development efforts.

d - The risks involved in implementation and operation of the proposed system.

5. Feasibility of Implementation - Proposed System

5.1 Implementation Risk - Proposed System

a - An estimate of the amount of innovation required in the system.

b - The existence (or lack) of development tools and aids.

c - The professional experience of the available manpower, with specific reference to the proposed system.

5.2 Development Process - Proposed System

a - The proposed development phases.

b - The proposed results for each phase.

c - The proposed main mile-stones.

5.2 Development Process - Proposed System cont'd

d - The proposed acceptance criteria for each result and each phase.

e - The proposed method to control modifications.

5.3 Preliminary Schedules - Proposed System

a - Schedule for phases.

b - Manpower schedule; types of man-power should be shown.

c - Computational resources schedule.

d - Support schedule - types of support should be shown.

e - Budget.

f - Product schedule.

5.4 Development Security - Proposed System

a - Catastrophe and accident prevention and recovery during development.

b - Disclosure risk for programs and test data.

c - Functions which must be secure.

d - Expected man-power turnover during development.

5.4 Development Security - Proposed System cont'd

e - External and internal audits during development.

f - Expected degree of responsibility of man-power.

5.5 Additional Clauses

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

REFERENCES

1. [BOE 76] Boehm, B.,

   "Software Engineering"; in IEEE Transactions on Computers C-25 (12); December 1976; pp. 1126-1241.

2. [BON 73] Bontempo, C.J.

   "Data Resource Management", Data Management, February 1973, pp. 33-37.

3. [GIL 74] Gildersleeve, T.R.,

   Data Processing Project Management, van Nostrand, Reinhold Co.; 1974.

4. [HTC 74] Hice, G.F., Turner, W.S., Cashwell, L.F.,

   System Development Methodology, North Holland; 1974

5. [IBM 75] Business Systems Planning: Information Systems Planning Guide; IBM Corp.; New York; August, 1975.

6. [MAR 73] Martin, J.

   Design of Man-Computer Dialogues; Prentice Hall, 1973.

7. [MAW 74] Mangold, E.R., Williams R.D.,
"Software Visibility and Management", in Boehm, B., Gaynor, E., Editors, TRW; 1974, pp. 201-265.

8. [MET 73] Metzger, P.W.
Managing a Programming Project, Prentice Hall, Inc., 1973

9. [PLA 72] Plagman, B.K., Altshuler, G.P.,
"A Data Directory/Dictionary System within the Context of an Integrated Corporate Data Base", AFIPS Fall Joint Computer Conference; vol. 21, Part II, 1972, pp. 1133-1140.

10. [ROA 74] Royce, W.W., Allen, D.J.,
"Software Requirements Analysis, Sizing and Costing", in Boehm, B., Gaynor, E., Editors; TRW, 1974; pp. 1.1 - 1.77.

11. [STA 77] v. Staa, A.,
Metodologias para o Desenvolvimento de Sistemas de Informação: Proposta de Desenvolvimento, Pontifícia Universidade Católica, Rio de Janeiro, Departamento de Informática, Internal Report; 1977.

12. [STA 79] v. Staa, A.,

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, Brazil, 1979.

13. [TEH 77] Teichroew, D., Hershey, H.E.,

"PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems", in *IEEE Transactions on Software Engineering* vol. SE-3; no. 1; January, 1977, pp. 41-49.

14. [WOO 74] Wooldridge, S.,

Computer Input Design, Petrocelli/Charter; 1974.

P Kyzik. 369/ #36

Printing Requisition/Graphic Services

Dept. No. 52075

Title or Description
CS 78-31 The Development Proposal - The First Step in Software System Construction

Date 11 Sept/79 Date Required ASAP Account 136-6040-41

Signature A. Tripodi Rocke Signing Authority

Department Computer Science Room 5182 Phone 3143

Delivery ☒ Mail □ Pick-up □ Via Stores □ Other

1. Please complete unshaded areas on form as applicable. (4 part no carbon required).
2. Distribute copies as follows: White, Canary and Pink—Printing, Arts Library or applicable Copy Centre Goldenrod—**Retain.**
3. On completion of order, pink copy will be returned with printed material. Canary copy will be costed and returned to requisitioner, **Retain as a record of your charges.**
4. Please direct enquiries, quoting requisition number, to Printing/Graphic Services, Extension 3451.

Reproduction Requirements ☒ Offset □ Signs/Repro's □ Xerox
Number of Pages 74 Number of Copies 55XXXXX

Type of Paper Stock ☒ Bond □ Book □ Cover □ Bristol □ Supplied

Paper Size ☒ 8½ x 11 □ 8½ x 14 □ 11 x 17

Paper Colour ☒ White □ Other Ink □ Black

Printing ☒ 1 Side □ 2 Sides Numbering to

Binding/Finishing Operations ☒ Collating □ Corner Stitching □ 3 Ring □ Tape □ Plastic Ring □ Perforating

Folding Finished Size 3 STAPLES Cutting Finished Size

Special Instructions
Please have the enclosed covers attached to the report, stapled on the top, middle and bottom, as the original is. Thnks.

Cost: Time/Materials		Fun.	Prod.Un. Cl.	Prod.Opr. No.	Mins.	Total
Signs/Repro's	1					
Camera	2					
Correcting & Masking Negatives	3					
Platemaking	4					
Printing	5	36-25				
Bindery	6		23-20 23-25			
57						
Sub. Total Time						
Sub. Total Materials						
Prov. Tax						
Total						

Film Qty Size	Plates Qty Size & Type
Paper Qty Size	Plastic Rings Qty Size
Outside Services	

The Development Proposal - The First Step in Software System Construction

D.D. Cowan *
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

A. von Staa
Departmento de Informatica
Pontificia Universidade Catolica
Rio de Janeiro - Brazil

Research Report CS-78-31
June,1978

Faculty of Mathematics

University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

The Development Proposal - The First Step in Software System Construction

D.D. Cowan *
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

A. von Staa
Departmento de Informatica
Pontificia Universidade Catolica
Rio de Janeiro - Brazil

Research Report CS-78-31
June,1978

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

THE DEVELOPMENT PROPOSAL
- THE FIRST STEP IN SOFTWARE
SYSTEM CONSTRUCTION

D.D. Cowan
Computer Science Department
University of Waterloo
Waterloo Ontario Canada.

A. von Staa
Departmento de Informatica
Pontificia Universidade Catolica
Rio de Janeiro - Brazil

ABSTRACT

When developing software systems using the top-down design and implementation approach, it is extremely important to state first the objectives of the development effort 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 called the Development Proposal. In this paper we show how to construct a development proposal, explain why it should be constructed in this manner and illustrate its review and evaluate its contents.

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 [MAW 74; MET 73] 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, 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 brush definition [ROA 74] since it is the first step in the refinement process.

We call this first document describing the software system "The Development Proposal" and we believe it is fundamental to the achievement of quality software products. This Development Proposal and its evaluation or acceptance criteria are the central issues of this paper.

Clearly 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 if the system is likely to satisfy its goals and constraints. For example, such questions as:

> Are the performance criteria likely to be acceptable?;
>
> Are the input and output formats acceptable to the users?;

and Are the operational constraints reasonable?

are important to the system and should be answered early in its development. If 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 development proposal is the first step. The next few paragraphs describe all the phases and basic steps in each phase. In many small or medium-scale software projects the effort required for some of the phases may be minimal. However, each phase should be recognized even if it is not implemented.

Problem Definition

The development proposal which is described in this paper is produced during this first phase.

Initial Planning

The development proposal has now been accepted and at this point, a development plan is produced which describes how to develop the system presented in the proposal. The plan may also state standards to be followed throughout the whole system's life cycle. In certain instances it might be appropriate to leave planning details of later phases or tasks until more knowledge has been acquired [GIL 74]. Even this "adaptive planning" must be planned, otherwise false expectations and frustration may develop.

Preliminary Design

In many projects a prototype or simulation model must be developed in order to understand some critical details of the software system. These details include such items as memory space, Preliminary Design cont'd.

execution time, and frequency of I/O requests. Preliminary design might be intertwined with the initial planning effort, or may even be placed before initial planning.

Functional Specification

During this phase a detailed specification of the input/output interface of the system is described. Reports, query capabilities, process control and other man-machine interactions are specified. In order to obtain this information an analysis of applicable portions of the existing system must be considered.

On many projects this phase is often called requirements analysis or problem definition [TEH 77]. The term requirements analysis is also sometimes used to describe a document similar to our "development proposal" [ROA 74]. To avoid confusion the term requirements analysis is not used in this paper.

Logical Specification

In the logical specification phase the program units and manual procedures are defined in terms of their interfaces and functions, but nothing is said about the mechanics of data processing. The inter-

Logical Specification cont'd.

face of the data base or file system is specified; such items as operations on the data base, and record layouts for all records which are interchanged with the rest of the system are defined.

Physical Specification

During this phase data representations and algorithms are selected.

Coding

At this point the programs are coded and tested. Both individual and integration tests are performed.

System Test

During this phase a thorough and pre-planned test is performed in order to verify whether the system behaves according to the specification.

Acceptance Test

A thorough and pre-planned test is performed to determine whether the system satisfies the user expectations and whether it performs adequately within its operational environment [NEW 74].

Site Test

During this phase the system is placed in operation, still within a monitored environment.

Operation

During the operation phase the system operates in a normal production environment. The system is also maintained, in order to adapt to new needs or fix errors.

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. Section 3 identifies the groups of people or entities involved in developing and using a computerized system. These entities are used in subsequent sections which present the details of the development proposal. In section 4 an outline of the method for constructing and reviewing the development proposal is presented. Section 5 describes the review criteria which are used to evaluate the development proposal. The final section reviews the entire concept of a development proposal. An outline of the development proposal by chapter, section and sub-section is presented in Appendix A. This table of contents 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,and constraints and this information must be acquired before any major planning effort is initiated. For example, there should be a clear comprehension 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. [MAW 74, ROA 74, METZ 73, HTC 74]. This background information will state major restrictions and mandatory requirements and allow predictions and estimates. All of this information can be used as an aid for generating alternatives during subsequent phases [ROA 74].

Software systems may be developed to replace existing systems or for entirely new applications. In the case of a replacement system, a basic but not necessarily detailed knowledge of the existing system must be acquired. This understanding is necessary in order to determine the feasibility of initiating the development, and to determine some of the goals which the replacement system must satisfy. The existing system should be described by a brief but clear and precise document which highlights its known problem areas.

The new system must also be described, since the whole development effort will use this description as a departure point. At this initial stage the description cannot be detailed, however, sufficient information must be presented to provide a clear, current picture of its goals, functions, operations, 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 if the results are feasible and if the two systems are substantially different.

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 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 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 design.

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, hence it determines the level of knowledge at the time the proposal was prepared. The review is not merely an acceptance criterion, but an explanation of the important portions of the proposal and 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 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 and its main purpose is to serve as an agreed commitment or contract between the users and system developers.

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. Currently, some new industrial systems are being formulated using the development proposal described in this paper [STA 77].

The current literature illustrates several different ways to produce the development proposal. 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 [IBM 75] 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, nature and geographical dispersion of the company. The business system planning method is very formalised 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 [TEH 77] proposes a documentation system and a formal mechanised 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 [ROA 74, HTC 74, MET 73] mention the need for a development proposal and give some details about its contents. However, no publication known to the authors presents a detailed description of a development proposal including its construction, contents and review methodology.

3. ENTITIES INVOLVED IN DEVELOPING A COMPUTERIZED SYSTEM

There are several groups of people affected by the development of a computerized system. Some groups participate in its development, while others may be classified as users. These various groups are called entities in this paper. This classification of the groups affected by a system is used in both the development proposal and the evaluation or review.

This section describes these groups according to their role in the system development and not according to their hierarchical or functional position within the institution sponsoring the system development. Clearly it is possible for a given individual to belong to more than one entity group.

3.1 Users

A user is anyone who is affected by the new system once it becomes operational. Obviously there are different classes of users. Some users are more severely affected by the new system than others.

This definition of user entity obviously includes both computer operators and input clerks. It includes also maintenance programmers since they are affected by the system once it becomes operationl. This particular group of users is called the operative user. It would be expected that the operative users have special training and possess adequate skills to perform their duties.

The definition of the user entity also shows that persons who prepare data for the system and those who receive results are users. This class of user is called direct user. In some cases direct users may also perform operational duties, as in an online system.

Examples of direct users are managers who receive results from the system, tellers who receive account information, tellers who create input for an online system, customers who receive bills and make payments, and programmers using a programming language. One can conclude from this list of examples that many different levels of training are required for members of the direct user group. Personal contact with some of the direct users is virtually impossible, but their needs still must be fulfilled by the system and hence one must state as precisely as possible the requirements of the direct users.

Owing to the lack of training of many of the individuals within the direct user group, the user system-interface must be carefully designed so as to

3.1 Users cont'd.

avoid exceptional or impossible requirements on the direct user [MAR 73]. An adequate interface may lead to frustration, and even to the abandonment of the new system.

Finally, there is another class of user, the indirect user, who is affected by the automated system. Such users are often subject to side effects of systems, without being aware of their involvement. For example, a person whose records are kept in a computerized system may be an indirect user. Such a class of user is of significant interest in any study on the impact of computers on society but they are beyond the scope of this paper. Other indirect users, such as passengers making a reservation on an airline interact with the system by means of a direct or operative user.

The basic difference between a direct user and an indirect user is that the indirect user does not have to understand the system he is using, hence needs neither training nor any special skill. On the other hand, the direct user must have at least some knowledge of the system, hence his level of training and skills affect the system interface. In many cases it will be difficult to tell the difference between the two classes of users. For example, receiving and paying a bill (direct use) requires almost no knowledge about the system; similarly, making an airline reservation (indirect use), normally requires some knowledge about the reservation process.

Certainly the most important user group is the direct user, since members of this group are the obvious victims or beneficiaries of the computerized system. Members of this group are often the most difficult to satisfy since they may have little or no understanding of what is to be accomplished with the computer system. Conversely, the developers of the system may not have an adequate understanding or appreciation of the needs of the direct user. Finally, the direct user cannot be described or defined by the system, since it is this group which actually defines the capabilities of the system. By contrast the operative user is at least partially described by the system through training and skill levels, although a significant effort should be expended in describing an operative user system interface [MAR 73, TOL 75, WOO 74, WOO 75].

3.2 The Co-ordinator

The co-ordinator entity is responsible for setting standards and ensuring adherence to these standards and controlling adherence to the development plans and methods. He is also responsible for effective user involvement during the development of the computerized system.

3.2 The Co-ordinator cont'd.

When a computerized system is developed with a team, there is a definite need for a development plan, otherwise development effort might be lost because of preparation of incompatible sub-products. Obviously the plan alone will do nothing - it must be continuously followed. It is the job of the co-ordinator to enforce this strict adherence to the plan. *

This characterization of the co-ordinator is similar, or even equivalent, to that of the conventional project leader. However in a large project, there might be several sub-projects, each of which is controlled by its own project leader. Furthermore the co-ordinator also undertakes tasks not usually associated with the project leader. These tasks include auditing and setting standards.

Finally the co-ordinator represents the institution for which the system is being developed. It should be noted at this point that direct users are not necessarily members of this institution. Also the implementor entity to be characterized later is frequently outside the institution.

* Planning and changes to the plan must be performed and agreed to by all of the participants or their representatives.

3.2 The Co-ordinator cont'd.

Although the co-ordinator is a representative of the institution, some of the project team members might be external. These external members typically perform tasks such as setting standards, design walk-throughs, and audit functions.

3.3 The Implementor

The implementor entity is responsible for the development and installation of the new system.

The implementor participates in the planning and usually controls adherence to those sections of the plan which are his direct responsibility. He is still subject to the standards set by the co-ordinator and to the control exerted by the co-ordinator.

Although it may be argued that the implementor and co-ordinator are the same person in many situations, we have chosen to distinguish between these two entities owing to the nature of the functions they each perform; the role of the co-ordinator is mainly management, whereas the role of the implementor is mainly production.

3.4 The Customer

The customer entity makes resources available to the project and decides upon its continuation or termination. The customer may be regarded as the general management of the institution for which the system is to be developed. In many cases the customer might delegate decision-making powers to the co-ordinator who then acts on behalf of the customer.

4. CONSTRUCTION AND REVIEW OF THE DEVELOPMENT PROPOSAL

Once the requirement for a new system has been determined then the first version of the development proposal should be produced, using the detailed guidelines presented in Appendix A. The first version of the proposal may not contain enough information to respond adequately to each item in the guidelines and so production of a final version is likely to be an iterative process.

Once the first version of a development proposal has been produced, it is reviewed by the co-ordinator and by the users or their representatives, and a review report is produced. Using the review report and the development proposal it is the co-ordinator'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 co-ordinator. This process continues until no more changes can be made, and both users and the co-ordinator 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 4.1 we show steps in the construction and review process for a development proposal.

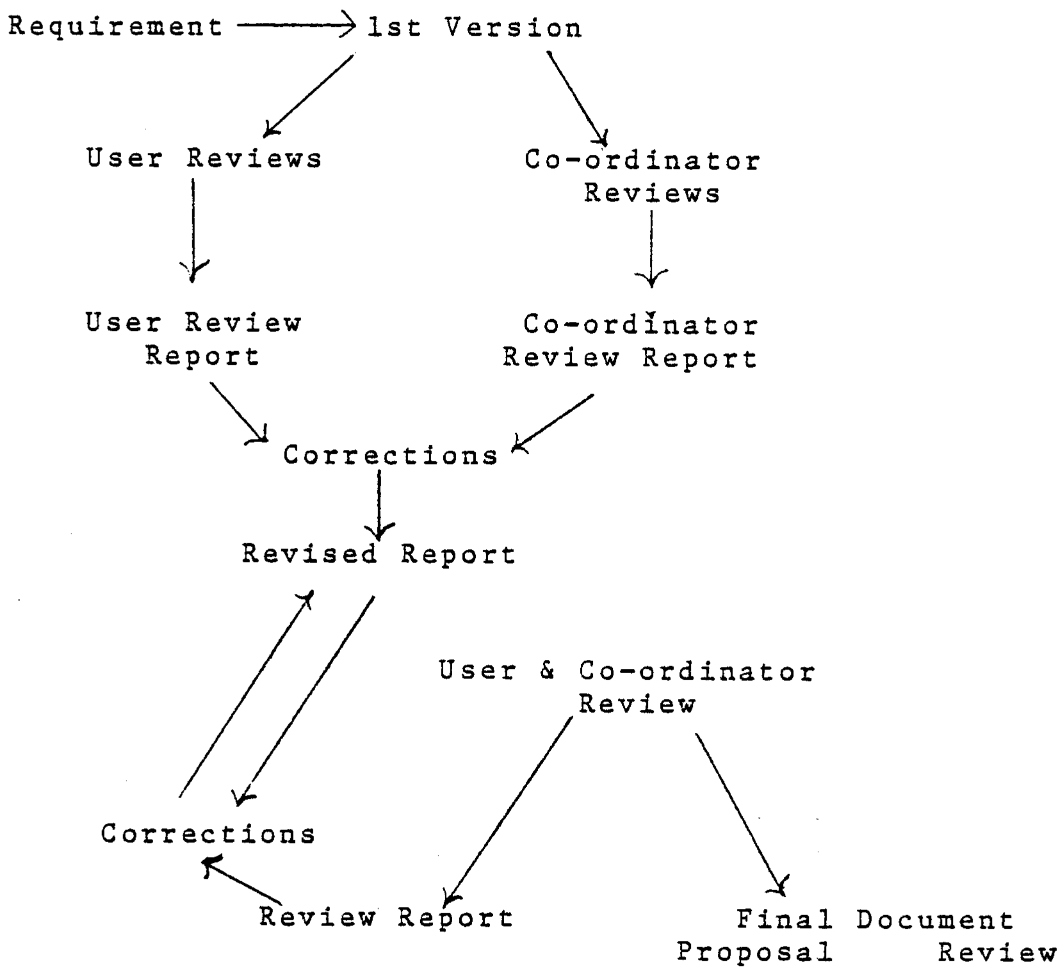

FIGURE 4.1 CONSTRUCTION AND REVIEW PROCESS

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:

(i) The user and co-ordinator cannot agree on a final proposed system because the users are unable to specify the requirements of the system.

(ii) The corrections to the development proposal negate previous corrections as no adequate record-keeping mechanism for changes has been instituted.

(iii) The use of excessive detail prevents completion of the proposal since the users and co-ordinator do not understand its purpose.

(iv) The system is developed independent of its final use; this is often the situation with systems developed as status symbols.

(v) Neither the users nor the co-ordinator 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 converge of the proposal but the examples are representative of most of the difficulties one is likely to encounter.

5. 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 system, and a review or evaluation of the adequacy of the current and proposed system.

The description method has been presented in a previous section and a typical table of contents is in Appendix A. The group preparing the development proposal amplifies each section and sub-section of the table of contents.

The review or evaluation method is presented in this section. As an evaluation procedure, the review must answer questions about the original description of the system to determine if 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 ansered 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 which 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 which attempt 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:

(i) The current system,

(ii) The proposed system,

(iii) The relation of the proposed system to the environment,

(iv) The operational feasibility of the proposed system,

and (v) The feasibility of developing the proposed system.

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

Several of the appraisal criteria overlap each other and hence contain some redundancy. Redundancy is a desirable property since it provides a means for checking results and hence it provides 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.

5.1 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 sub-divided into (a) description, (b) current organization, (c) necessity, (d) complexity, (e) security.

5.1.1 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.

5.1.1 Description of the Current System cont'd.

Evaluation requires information about: (a) the tasks performed by the current system - both manual and computerized; (b) the objective of these tasks; (c) the data flow between tasks; (d) the service requests which may occur and which tasks are triggered by these requests; (e) the origin of the service requests; (f) the periodicity, or volume per time unit, of service requests; (g) the results required for each type of service request; (h) the external inputs per task; (i) the data maintained by the current system; (j) the storage media such as ledgers, file cabinets, tapes and disks; (k) the expected and real elapsed time from service request reception to output distribution; and (l) the number of service requests which are not satisfied, and main reasons for this lack of satisfaction.

5.1.2 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 ncessary or desirable. In many cases the

5.1.2 Current Organization cont'd.

observed need for a new system stems from an inadequate organization. Installing a computerized system in such a case will not alleviate these organizational difficulties and in fact it 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: (a) the tasks performed by the present system; (b) the data flow between tasks; (c) the controls which are imposed on the operation of the present system; (d) the operational difficulties of the present system; (e) the volume of stored data; (f) the storage media used for the data; (g) any access and retrieval difficulties; (h) the number of operational users involved in the operation of a task; (i) the number and complexity of processing steps for each service request; (j) the number

5.1.2 Current Organization cont'd.

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 man-hours; (k) the number of service requests which have not been properly processed and the reasons; (l) the real and expected elapsed time between service requests arriving, and output distribution; (m) the number, size (number of data items) and complexity of forms used by the present system; (n) the operating costs of the present system; (o) the volume of service requests; and (p) examples of each class of user emphasizing training and skills.

5.1.3 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 value of the current system features to both the direct and indirect user, 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 which is really operating in a satisfactory manner.

5.1.4 Current Systems Complexity cont'd.

The evaluation requires information about (a) the tasks performed by the current system; (b) the data flow in the current system; (c) the effect of data which is imprecise, incomplete and not current on the tasks of the system; (d) 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 inter-dependancy of the stored data - should be examined; (e) the number of forms required; (f) the volume of services required; (g) the expected and real elapsed time from service request arrival to output distribution; (h) the number of steps required to produce desired outputs per service request; and (i) the operative user - the training and skills required.

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

5.2 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 if there are any requirements being placed on the proposed system which are either superfluous or not realistic.

The appraisal criteria for the proposed system may be sub-divided into:

(a) Definition of the proposed system;
(b) Definite restrictions on the proposed system;
(c) The life expectancy of the proposed system;
(d) The complexity of the proposed system;
(e) The complexity of the solution to the problem relative to the task to be performed;
(f) Security of the proposed system;
and (g) Benefits of the proposed system.

5.2.1 Definition of 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 enough information about the system, an experienced designer should be able to produce an intuitive functional specification which shows how the proposed system will operate. Obviously, this preliminary concept is likely to be quite different from the system design which 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 which will have to be designed [ROA 74]. 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: (a) a statement of the automatic and manual processes contained within the proposed system; (b) the description of these processes; (c) the data flow between the processes; (d) the necessary data to activate each individual process; (e) the output for each process; (f) the data maintained by the processes, specifically the type of data but not the implementation details; (g) the service requests and their descriptions; (h) the origin of service requests; (i) the frequency, or volume per time unit, of each service request; (j) the expected elapsed time or response time from service request arrival to output distribution; (k) the

5.2.1 Definition of Proposed System cont'd.

responsibility for input data; (l) the responsibility for storing and maintaining data; (m) the need for the outputs; (n) the security requirements for input data; (o) the security requirements for output data; (p) 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 online files, dismountable files, filing cabinets and ledgers; and (q) the expected volumes of inputs and outputs, data bases and service requests.

5.2.2 Definite Restrictions

This criterion determines the validity of any pre-imposed design and implementation alternatives.

Pre-imposed 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 pre-defined and cannot be freely chosen during design and implementation. Since such standards or pre-imposed alternatives are based frequently on current conditions and may change in the future, they should be reviewed and restated explicitly for each new project.

5.2.2 Definite Restrictions cont'd.

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: (a) restriction description, (b) justification for the restriction.

Typical restrictive items are: (i) hardware; (ii) portability; (iii) operating systems; (iv) software support systems; (v) languages; (vi) input/ output equipment; (vii) forms; (viii) labels, density, and layout for each file; (ix) development procedures; (x) documentation procedures; (xi) programming style; (xii) system types available - such as online systems, centralized systems, and networks; (xiii) maintenance requirements; and (xiv) 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.

5.2.3 Proposed Organization

This criterion measures the impact a proposed computer system will have on the user organization

5.2.3 Proposed Organization cont'd.

and 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 user must be made aware initially of his/her responsibility in both the areas of implementation and operation of the system.

The evaluation of the proposed organization should consider: (a) the processes provided by the proposed system; (b) the data flow; (c) the groups responsible for producing input data; (d) the groups responsible for storing and maintaining data; (e) the outputs of the system and the destination of these outputs; (f) the expected use to be made of these outputs; (g) the storage media for the data; (h) the mechanisms to gather and transcribe data into machine-readable form; (i) the number and skills of all operations personnel; (j) the number and skills of all maintenance personnel; (k) the controls of the system - including security and quality control; (l) the expected elapsed time between input and receipt of output - the so-called response time; (m) any equipment restrictions;

5.2.3 Proposed Organization cont'd.

(n) any software restrictions; and (o) programming restrictions.

5.2.4 Life Expectancy

This criterion measures the expected life-time 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 should be enhanced, in advance of a crisis. Furthermore, it is possible to enforce expansion criteria from the beginning of a system and thus hopefully reduce costs of unexpected expansion.

An evaluation of life expectancy should consider:

(a) the expansion policy of the user entity;

(b) the integration and centralization policies of the user entity; (c) the expected growth of the data base; (d) the expected growth of service requests; (e) the expected growth of future additions to the systems; (f) the current volume of the data base; and (g) the current volume of the service requests.

5.2.5 Complexity of the Proposed System

This criterion measures the operational difficulty which 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, while others will be caused by the naivety of the system design. Stating the operational difficulties acts as a design control and also is an indicator of the necessity of a more careful analysis and system design to be performed during later phases.

An evaluation of complexity should consider: (a) the processes involved; (b) the data flow; (c) the

5.2.5 Complexity of the Proposed System cont'd.

origin and responsibilities for data; (d) the responsibility for storing and maintaining data; (e) the data gathering and data transcription procedures; (f) the controls; (g) the output distribution; (h) the volume of service requests, (i) the expected maximum elapsed time between service request arrival and output distribution; (j) the estimated volume of processing per service request; (k) the number of steps required to handle a service request; (l) the ability of the system to handle data which is imprecise, incomplete, and not current; (m) the storage media for data; (n) security against accidents; and (o) security against corruption such as the misuse or wilful destruction of data or programmes.

5.2.6 Simplicity of Solution

This criterion measures whether certain functions, outputs and service requests are really necessary.

The main purpose of this criterion is to verify whether the proposed 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

5.2.6 Simplicity of Solution cont'd.

as in sub-section 5.1.3. One should also include: (a) the expected use of outputs; and (b) expected users of system output.

5.2.7 Security of the Proposed System

This criterion measures 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 which 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: (a) the processes involved; (b) the data flow; (c) the controls; (d) the responsibilities for input data; (e) the responsibility for storing and maintaining data; (f) the ability of processes to accept data which is imprecise, incomplete or not current; (g) the impact of loss or destruction of data; (h) the possible advantages which might accrue to an unauthorized user who has illegally obtained data; (i) the proposed accident prevention measures; (j) the proposed reconstruction routines; (k) the expected volumes of stored and maintained data; (l) the storage media for data; (m) the expected update frequency for data; (n) the number and skills of operators; and (o) the number and skills of maintenance programmers.

5.2.8 Benefits of the Proposed System

This criterion measures the benefits that the proposed system will provide to the user entity and to the customer entiry.

One of the first uses of the development proposal is to perform a cost analysis of the proposed system. This means that the benefits also have to be stated. In recording the expected benefits of the proposed system we obtain a list of the user expectations.

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

The benefit evaluation examines: (a) the operational difficulties of the present system; (b) the operational difficulties of the proposed system; (c) the security sub-system of the present system; (d) the security sub-system of the proposed system; (e) the number of unsatisfied service requests in the present system; (f) the expected volume of service requests; (g) the expected growth in service requests; (h) the estimated changes in personnel; (i) the dependancy of the user on availability of equipment and external people; and (j) the list of proposed benefits - this list should show not only the benefit but should also give a brief description of the importance of such a benefit, without quantifying it.

5.3 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 (a) singularity, (b) integrability (c) use by other user groups and (d) global benefits and priority.

5.3.1 Singularity

This criterion determines whether the proposed system is unique or whether it is 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: (a) the existence of similar systems within the users' institution; (b) the existence of a similar system, on the software market; (c) the estimated effort required to adapt systems which already exist; (d) the difficulty of maintaining a system which was not developed in the home installation; (e) the satisfaction of user needs with an acquired system; and (f) the ability to generalize the proposed system to incorporate more users.

5.3.2 Integration of the Proposed System with Other Systems

This criterion measures whether data can be interchanged between the proposed system and other systems being developed, or in operation.

5.3.2 Integration of the Proposed System with Other Systems cont'd.

The current trend is toward overall top-down design of integrated systems [IBM 75], 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 from the 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 which are required for integrated systems will still be difficult to implement if the operations cross system boundaries.

The evaluation requires knowledge about: (a) the existence of a data directory/dictionary [BON 73, PLA 72]; (b) the existence of systems sharing some input data; (c) the existence of systems with common user sub-groups; and (d) the estimated effort to achieve data-commonality for systems sharing some input or maintained data.

5.3.3 The Ability of the System to be Used by Other Users

This criterion attempts to determine the existence of potential user groups which have not yet been recognized.

In some organizations copies of a system may be used by different user groups, such a situation would occur 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 is quite difficult and must be known in advance.

The evaluation of this criterion requires knowledge of: (a) the existence of user groups with similar job goals; (b) the effort required to adapt or customize the system to each of the user groups; and (c) the difficulty of communicating with different user groups.

5.3.4 Global Benefits and Priorities

This criterion determines the priority of the proposed system with regard to other systems which are under development or which are being considered for development.

Most organizations have a limited capacity to develop new systems; budgetary and man-power 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 man-power, 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: (a) the benefits of the proposed system for the user; (b) the relative importance of the proposed system for the organization; (d) the expected benefits for the organization; (e) the impact on other projects; (f) the efficiency and effectiveness of the present system; and (g) the risks involved in implementation and operation of the proposed system.

5.4 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 wilful misuse are often the result of a system which 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 [MAR 73].

This appraisal class contains the following criteria: (a) the operational cost; (b) the computation restrictions; (c) the operational environment; and (d) the operational security.

5.4.1 Operational Cost

This criterion measures the cost restrictions which apply to a system once it becomes operational.

The cost of operating a system is related to its complexity, its processing volume, and its

5.4.1 Operational Cost cont'd.

required response times, as well as 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 a knowledge of: (a) the number and skills of the persons operating and maintaining the system; (b) the proposed security sub-system; (c) the proposed data acquisition sub-systems; (d) the proposed data maintenance sub-systems; (e) the volume of stored and maintained data; (f) the processing volume; (g) the response time restrictions; (h) the life expectancy of the proposed system; (i) the growth expectancy of the proposed system; (j) the estimated probability of not achieving the goals desired of the system; (k) the estimated equipment cost; (l) the estimated man-power cost; (m) the estimated maintenance cost; and (n) other costs - such as those for forms, materials and rent.

5.4.1 Operational Cost cont'd.

Some cost figures in this list depend upon decisions which 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 to 30% [BOE 73]) in present day systems, and so their total effect on these cost figures should not be too significant.

5.4.2 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: (a) the expected response time from

5.4.2 Computation Restrictions cont'd.

service request arrival to output distribution; (b) the number of functions required to serve a request; (c) the estimated response times per function; (d) the storage requirements per function; (e) the estimated storage requirements for the data base by storage media; (f) the complexity of the proposed system; (g) the processing volume per service request; (h) the volume of service requests per time unit; (i) the functions of the proposed system; (j) the data flow of the proposed system; (k) the data acquisition sub-system; (l) the security sub-system; (m) the data maintenance sub-system; and (n) the estimated equipment cost.

5.4.3 The Operational Environment

This criterion evaluates the constraints which 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 [MAR 73, WOO 74].

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 can be terse and allow a certain amount of creativity, or it may be verbose and presented in a "cookbook" manner.

Evaluation of this criterion requires a knowledge of: (a) the number and skills of the operators; (b) the number and skills of the maintenance programmers; (c) the skills of direct users; (d) the expected number of replacement personnel to be

5.4.3 The Operational Environment cont'd.

hired in a given time period; (e) the responsibility level of the operative user; (j) the contact that operative or direct users will make with the other users; (g) the security sub-system; (h) the complexity of the proposed system; (i) the functions of the proposed system; and (j) the data flow.

5.4.4 Operational Security

This criterion measures the suitability of the proposed security sub-system.

Data and programs are valuable resources. Their disclosure to others might cause losses, law suits 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: (a) the dependence of the functions on complete, precise and current data; (b) the functions of the proposed system; (c) the data flow; (d) the importance of the system to the

5.4.4 Operational Security cont'd

user; (e) the required privacy and confidentiality of data manipulated by the proposed system; (f) the estimated losses when data or programs deteriorate or are destroyed; (g) the estimated losses if data or programs are disclosed to unauthorized persons; (h) the existence of measures to prevent catastrophes; (i) the existence of internal and external auditing functions; (l) recovery procedures; and (m) emergency operation procedures - such as offline execution of the system when the online system is not operating.

5.5 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 re-evaluated at appropriate points in time [GIL 74].

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: (a) technical feasibility; (b) schedule feasibility; (c) man-power feasibility; (d) support feasibility; (e) financial feasibility; (f) development security; and (g) additional clauses.

5.5.1 Technical Feasibility

This criteria measures the technical difficulty of implementing the proposed computer system.

The evaluation requires knowledge about: (a) the function of the proposed system; (b) the data flow; (c) the security sub-system; (d) the expected elapsed time per service request; (e) the expected processing volume; (f) the expected volume of service request; (g) the number of functions per service request;

5.5.1 Technical Feasibility cont'd.

(h) expected response time per function; (i) the data acquisition sub-system; (j) the data maintenance sub-system; (k) the volume and storage media of maintained data; (l) the estimated complexity of the system; (m) the estimated amount of innovation required in the system; and (n) 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.

5.5.2 Schedule Feasibility

This criterion measures the adequacy of the proposed schedule for the development of the system.

The evaluation should consider: (a) the phases of the development schedule - names, descriptions and products; (b) the estimated duration of each phase; (c) the estimated technical and administrative man-power required for each phase; (d) the

5.5.2 Schedule Feasibility cont'd.

estimated computational resources for each phase; (e) the estimated requirements for other resources such as special equipment, materials, rentals, documentation; (f) the complexity of the proposed system; (g) the estimated amount of technological innovation required; (h) the possible difficulties which might arise during each phase - that is, the risk factors; and (i) the acceptance for each phase.

5.5.3 Man-power Feasibility

This criterion measures the adequacy of the estimated man-power requirements.

The evaluation requires information about: (a) the phases of the proposed development plan; (b) the functions of the proposed system; (c) the number and skills of technical man-power required; (d) the number and skills of clerical man-power required; (e) the complexity of the proposed system; (f) the estimated amount of technological innovation required; (g) the expected professional experience of the man-power required; (h) the expected professional skill for each phase; (i) the development security sub-system; and (j) the operational security sub-system.

5.5.4 Support Feasibility

This criterion measures the adequacy of the support which would be provided during the development of the system.

The evaluation should consider: (a) the phases of the proposed development plan; (b) the complexity of the proposed system; (c) the estimated amount of technological innovation; (d) the expected professional experience of the man-power required; (e) the estimated computational support requirements for each phase; (f) the estimated administrative support requirements for each phase; and (g) the other estimated support requirements for each phase.

5.5.5 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 about: (a) the phases of the proposed development plan; (b) the estimated cost of each phase; (c) the estimated man-power for each phase; (d) the estimated support requirements for each phase; (e) the estimated duration of each phase; (f) the acceptance criteria for each phase; and (g) the payment schedule and its links to the development plan.

5.5.6 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: (a) the operational security sub-system; (b) the internal and external operational audits; (c) the development man-power; (d) the responsibility level of the development man-power; (e) the disclosure risk for programs and test data; (f) the expected personnel turnover during development; (g) the functions which are critical to security; (h) the functions which are critical when modified in a transparent manner, for example, does a function behave adequately but contains non-specified extensions which might benefit others?; (i) the existence

5.5.6 Development Security cont'd.

of recovery procedures for the development; (j) the existence of procedures for preventing catastrophe during development; and (k) the existence of change control procedures.

5.5.7 Additional Clauses

This criterion measures the necessity of any additional contractual clause.

Additional contractual clauses 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, man-power selection, and specific 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.

6. EPILOGUE

The utility and necessity of the development proposal has been discussed and an attempt has been made to define its contents and review process. A guideline for the contents of this proposal is outlined in the Appendix. This guideline 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 review of another guideline [STA 77]. This former development proposal guideline has been used to define small (less than a man-year) application systems. Several minor flaws have been observed and, hopefully, eliminated in this current version. A major assessment of this current version is not yet possible, however, it will be evaluated by applying this version of the development proposal to another set of small and medium system definitions.

APPENDIX A

The body of this paper has described the reasons for a development proposal and its method of construction. This Appendix presents a table of contents for a sample development proposal. The table of contents gives titles for each chapter of the report and in most cases shows section and sub-section headings.

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

The sub-section 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 adequately answers the specific queries.

It should be noted that whenever the pronoun "who" is used in this Appendix it denotes a man-power position rather than a specific individual.

ABSTRACT

TABLE OF CONTENTS

1. Summary

2. Present System

2.1 Present System - Tasks

a - Tasks performed by the present system.

b - Objectives of these tasks.

c - Data flow between tasks.

d - Data required to activate tasks.

e - Volume of input data per task.

f - Volume of results per task.

g - Number of operational steps (maximum/minimum) required per task, including sorting, aggregating, computing and similar operations.

h - Nature of the present system - is the present system centralized, decentralized, computerized, partially automated or manual?

i - The importance of the results produced by the system.

2.2 Present System - Service Requests

a - What are the service requests?

b - Who originates the service requests, a person, an event or a time dependancy?

c - How are service requests originated?

d - The frequency of occurrence.

e - Tasks which must be performed to provide the required services.

2.2 Present System - Service Requests

f - Expected results per service request.

g - Expected and real response time from service request arrival to output distribution.

h - What is the number (volume or frequency) of service requests left unsatisfied and the major reasons for lack of service?

2.3 Present System - Data

a - Who collects and prepares data?

b - Who is responsible for the maintenance of data?

c - What are the storage media for data maintained by the system?

d - What are the access or retrieval mechanisms and their inherent difficulties?

e - What is the frequency of occurrence of incorrect, incomplete or non-current data?

f - How well does a task function using incomplete, imprecise or non-current data?

g - Description of the forms used by the present system.

2.4 Present System - Controls

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

b - What methods are used to determine whether the results are complete?

2.2 Present System - Service Requests

f - Expected results per service request.

g - Expected and real response time from service request arrival to output distribution.

h - What is the number (volume or frequency) of service requests left unsatisfied and the major reasons for lack of service?

2.3 Present System - Data

a - Who collects and prepares data?

b - Who is responsible for the maintenance of data?

c - What are the storage media for data maintained by the system?

d - What are the access or retrieval mechanisms and their inherent difficulties?

e - What is the frequency of occurrence of incorrect, incomplete or non-current data?

f - How well does a task function using incomplete, imprecise or non-current data?

g - Description of the forms used by the present system.

2.4 Present System - Controls

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

b - What methods are used to determine whether the results are complete?

2.4 Present System - Controls

c - What methods are used to determine the currency of the data?

d - What are the correction and recovery procedures?

2.5 Present System - Security

a - What are the measures for prevention of a catastrophe?

b - How confidential is the data manipulated by the system?

c - What is the degree of privacy for maintained data?

d - Who has access to the data under different conditions?

e - What are the existing recovery mechanisms?

f - What are the possible consequences (losses, fire, and other problems) of unauthorized access?

g - What are the possible consequences of unauthorized modification to the data?

2.6 User Entity

a - Operative User

i. - skills and training.

ii. - training time.

iii. - rate of attrition.

iv. - organization.

v. - categories of operative users.

2.6 User Entity

b - Direct User

i. - skills and training as related to the present system.

ii. - opportunity to train·

iii. - rate of attrition.

iv. - examples of categories and uses.

c - Indirect User

i. - access to the system.

ii. - examples of uses.

2.7 Importance of the Present System

a - How important is the present system in achieving the objectives of the user entity?

b - What losses occur as a consequence of inadequacies in the present system?

c - How important is the present system in achieving the objectives of the customer entity?

3. Proposed System

3.1 Proposed System - Functions

a - Functions (manual or computerized) to be performed by the proposed system.

b - Objectives of these functions.

c - Data flow between functions.

d - Necessary data to activate functions.

e - Estimated volume of input data per function.

f - Estimated volume of results per function.

g - Estimated processing volume per function.

h - Importance of the results to be produced by the proposed system.

i - Nature of the proposed system (centralized, decentralized, online, batch, or distributed).

3.2 Proposed System - Service Requests

a - What are the proposed service requests?

b - Who originates these service requests?

c - How are service requests originated?

d - Frequency of occurrence.

e - When are service requests originated?

f - Expected results per service request.

g - Expected response time per service request.

h - Controls applying to service requests.

3.3 Proposed System - Data

a - Who will gather and prepare data?

b - Who will be responsible for maintenance of the data?

3.3 Proposed System - Data

c - What are the proposed storage media?

d - What is the expected data volume at installation and what is the estimated capacity?

e - How well will functions perform with data which is incomplete, imprecise or not current?

3.4 Proposed System - Controls

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

b - What methods are used to determine whether the results are correct and complete?

c - What methods are used to determine the currency of the data?

d - What are the proposed correction and recovery procedures?

3.5 Proposed System - Security

a - What are the proposed measures for prevention of a catastrophe or accident?

b - How confidential is the data to be manipulated by the proposed system?

c - What is the degree of privacy for the maintained data?

3.5 Proposed System - Security

d - Who can read or modify the data?

e - What are the planned recovery mechanisms?

f - What are the possible consequences of losses?

g - What are the expected reasons (benefits, or vengeance) of unauthorized access?

h - Emergency operation procedures.

3.6 Proposed System - User Entity

a - Operative User

i. - expected skills and training.
ii. - expected time to train.
iii. - expected rate of attrition.
iv. - proposed organization.
v. - examples of categories.

b - Direct User

i. - expected skills and training in the use of the proposed system.
ii. - opportunity to train.
iii. - expected rate of attrition.
iv. - example of categories.

c - Indirect User

i. - proposed access to the system.
ii. - examples of uses.

3.7 Expectations

a - Date of initial operation.

b - Minimal cost to develop.

c - Minimal cost to operate over a period of time.

3.7 Expectations cont'd.

d - Definite equipment requirements.

e - Expected data growth.

f - Expected growth of service requests.

g - Expected growth of functions.

h - What are the dependencies to be tolerated (portability, languages, operating systems)?

i - Benefits of the proposed system - statement and justification.

4. Relation of Proposed System to Environment

4.1 Uniqueness of Proposed System

a - A brief overview of existing similar systems within the installation and on the market.

b - The effort required to adapt existing systems to fulfill user needs.

c - The expected difficulty of maintaining an adapted system.

d - The expected user satisfaction using an adapted system.

e - The potential additional users of the proposed system.

4.2 Integration of Proposed System

a - Does a data directory/dictionary system exist?

b - A brief overview of other systems which might or will share data.

c - A brief overview of other systems which might share direct or indirect users.

d - The expected effort to achieve data commonality for systems sharing data.

4.3 User Expansion

a - The existence of user groups with similar job goals.

b - The effort required to encompass these additional user groups.

c - The difficulty of communicating with user groups.

4.4 Global Benefits and Priorities

a - The relative importance of the proposed system for the organization.

b - The expected benefits for the organization - description and justification.

c - The impact of the development of the proposed system on other ongoing development efforts.

d - The risks involved in implementation and operation of the proposed system.

5. Feasibility of Implementation - Proposed System

5.1 Implementation Risk - Proposed System

a - An estimate of the amount of innovation required in the system.

b - The existence (or lack) of development tools and aids.

c - The professional experience of the available manpower, with specific reference to the proposed system.

5.2 Development Process - Proposed System

a - The Proposed development Phases.

b - The proposed results for each phase.

c - The proposed main mile-stones.

d - The proposed acceptance criteria for each result and each phase.

e - The proposed method to control modifications.

5.3 Preliminary Schedules - Proposed System

a - Schedule for phases.

b - Manpower schedule; types of man-power should be shown.

c - Computational resources schedule.

d - Support schedule - types of support should be shown.

e - Budget.

f - Product schedule.

5.4 Development Security - Proposed System

a - Catastrophe and accident prevention and recovery during development.

b - Disclosure risk for programs and test data.

c - Functions which must be secure.

d - Expected man-power turnover during development.

e - External and internal audits during development.

f - Expected degree of responsibility of man-power.

5.5 Additional Clauses

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

REFERENCES

[BOG 74] Boehm, B., Gayuor, E., Editors
Symposium on Reliable, Cost-Effective, Secure Software, TRW Systems Group; 1974.

[MAW 74] Mangold, E.R., Williams, R.D.,
"Software Visibility and Management";
in Boehm, B., Gaynor, E., Editors, TRW; 1974
pp. 201-265.

[MET 73] Metzger, P.W.,
Managing a Programming Project,
Prentice Hall, Inc., 1973.

[ROA 74] Royes, W.W., Alley, D.J.,
"Software Requirements Analysis, Sizing and Costing", in Boehm, B., Gaynor, E., Editors;
TRW; 1974; pp. 1.1 - 1.77.

[GIL 74] Gildersleeve, T.R.,
Data Processing Project Managerment,
van Nostrand, Reinhold Co.; 1974.

[MAR 73] Martin, J.,
Design of Man-Computer Dialogues;
Prentice Hall, 1973.

[TOL 75] Tameski, E.A., Lazarus, H.,
People Oriented Computer Systems;
van Nostrand Reinhold Co.; 1975.

[WOO 74] Wooldridge, S.,
Computer Input Design,
Petrocelli/Charter; 1974.

[WOO 75] Wooldridge, S.,
Computer Output Design,
Petrocelli/Charter; 1975.

[IBM 75] Business Systems Planning: Information Systems Planning Guide,
IBM Corp.; New York; August, 1975.

[TEH 77] Teichroew, D., Hershey, H.E.,
"PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems",
in *IEEE Transactions on Software Engineering*,
Vol. SE-3; No. 1; January, 1977, pp. 41 - 49.

[HTC 74] Hice, G.F., Turner, W.S., Cashwell, L.F.,
System Development Methodology,
North Holland; 1974.

[NEW 74] Nelson, E.C., Wilde, G.R.,
"Software Reliability, Verification and Validation". In Boehm, B., Gaynor, E., Editors, TRW, 1974, pp. 5.1 - 5.99.

[STA 77] v. Staa, A.,
Metodologies para o Desenvoloimento de Sistemas de Informacao: Proposta de Pesenvoloimento, Pontificia Universidade Catolica, Rio de Janeiro, Departemento de Informatica, Internal Report; 1977.

[BON 73] Bontempo, C.J.,
"Data, Resource Management",
Data Management, February 1973, pp. 33 - 37.

[PLA 72] Playman, B.K., Altshuler, G.P.,
"A Data Directory/Dictionary System within the Context of an Integrated Corporate Data Base",
AFIPS Fall Joint Computer Conference;
Vol. 21, Part II; 1972; pp. 1133 - 1140.

[BOE 73] Boehm, B.,
"The High Cost of Software".