The technical specification—key to management control of computer programming

by BURT H. LIEBOWITZ
Bellcomm, Incorporated
Washington, D. C.

INTRODUCTION
If one considers the life cycle of a system as described in the paper by Ratynski, it becomes evident that effective management is dependent upon the presence of detailed specifications. In general, a specification describes what a product should be so that someone or some group can design and/or build it. In the previous paper the concept of a two-part specification was introduced—the first part describing the technical requirements for the item, the second part describing the actual configuration of the completed item. In this paper we will consider the content of the specification and its application to the management control of the computer programming process. Particular emphasis will be placed on part 1 of the spec. In doing so we will pose and answer two questions: (1) why is the part 1 spec so critical to management control? and (2) why, if so important, have so many programming efforts been characterized by its absence?

To answer the first question, let us consider the objective of and the steps required for management control.

The objective of management control is to assure that the resultant program does the job for which it was intended, is ready when needed, and is produced within expected costs. To achieve management control the managing group must: develop a standard which describes expected performance, evaluate actual performance against this standard, and take corrective action when undesired deviations from the standard occur. Performance in this context covers both the performance of the group developing the program and the performance of the program itself.

For computer programming, the cycle formed by these steps looks something like that shown in Figure 1. The standard is expected costs, estimated schedules, and desired technical characteristics. Evaluation is accomplished by means of reviews and tests. Corrective action is accomplished by allocations of resources, clarification of the standard and, when necessary, changes to the standard.

The part 1 specification is an important segment of the management standard. It documents the technical requirements for the computer program, thus providing both the basic "design to" guide for the programming contractor and the standard by which the user can evaluate the resultant program. Without it the program developer must rely on word of mouth and informal notes and the user has no way of determining the validity of the evolving program. Thus there is no firm basis for management control.

To answer the second question, we must consider several factors, some common to both hardware and computer program development, others peculiar to computer program developments.

In the former category we find in some cases that lack of time prevents an adequate definition of requirements. Also, in some cases the nature of the effort is such that total requirements are not known prior to some experimentation with the system as (partially) built.

In computer programming we are faced with one other factor—the flexibility of the programmable digital computer. A somewhat exaggerated view of this capability had led unwary managers to the belief that, since changes to a finished program can be easily made, the time and costs to define requirements, and the need to exercise control once they have been

*For convenience in this paper, the managing group will be referred to as the "user" or "using agency"; the program developer will be referred to as the "contractor."
Changes or additions (from further system analysis)

The management standard

Technical characteristics; expected costs; schedules

Evaluation by means of tests, reviews

Computer program production

Outputs of the programming process:
  - Design documents;
  - Test results;
  - Actual costs;
  - Milestone completion dates

Change or clarify the standard

No changes required

Change the program; redeploy resources

released, can be avoided in computer programming. Another difficulty is caused by the use of computers to provide flexibility in systems which operate in a changing environment. These systems are designed so that computer programs will absorb the major portion of expected changes. This allows firm definition of hardware requirements early in the development cycle of the system but makes it difficult to define the computer program requirements at the same time.

The impossibility under realistic conditions of establishing complete and definitive requirements in the programming process should not, however, be interpreted as a hopeless impasse to management control. On the contrary, if we accept as the general rule that the definition of requirements is an evolving process, we must conclude that a major function of management control is to insure that the development of the part 1 spec is an orderly one. This can be accomplished by providing mechanisms for the development of the spec and then for controlling changes to the spec as the requirements become better defined. This is actually what configuration management is all about as discussed in the paper by Searle and Neil.

The rest of this paper will discuss the content of the first part of the specification, some methods for developing it in face of the realities mentioned above, and its relationship to the second part of the spec and other management control tools.

We begin with a brief description of what the first part of a computer program specification should contain.

The technical specification (Part 1)

Part 1 of the specification results from an analysis of the role the computer is to play in the system;
it is a translation of user needs to the language of computer systems analysts. The specification is developed in the definition phase prior to extensive design of the program.

Content of the specification

The computer program specification contains performance requirements, interface requirements, design requirements, and test requirements.

The performance requirements describe what the program is to do and the required data base characteristics. For each major function of the program the specification describes the type of data inputs, the required computer processing and the required outputs. The interface requirements describe the content and format of data transferred between the computer program and other computer programs and equipments. The design requirements delineate items, not directly related to the desired performance of the program, which constrain the design of the program, e.g., requirements for modularity and expandability. The test requirements define the types of tests needed to verify that the resultant computer program satisfies the performance and design requirements contained in the specification. Test requirements are included in the specification to allow the contractor sufficient time to adequately plan for running the required tests and to facilitate management control of the testing process.

An outline of a model format for the first part of the specification is given in Figure 2. A description of a model spec is given in the Appendix of this paper. The format was developed with two things in mind: (1) to provide a convenient framework for the documentation of computer program requirements, and (2) to conform as much as possible to the format already in use for hardware specifications. The format and model were produced as part of an effort, undertaken on behalf of NASA’s Apollo Program Office, to apply configuration management to computer programming. This effort was closely coordinated with a similar one undertaken by the USAF’s Electronic Systems Division.

Development of the specification

Ideally, the user would produce a complete set of requirements prior to the design efforts of the contractor. With adaption of the phased system acquisition concept as discussed by Ratynski, this ideal can be more closely approached than has been in the past. In this approach the interface, performance, design and test requirements would be developed in the definition phase prior to the contractor’s design efforts in the acquisition phase (see Figure 3). However, in many cases for the reasons discussed earlier in this paper, some overlap of requirements analysis with design must be allowed if the program is to be available when it is needed.

An orderly development of requirements can still be achieved in these cases. The requirements can be built up in several ways depending on exact circumstances. In some cases the functions of what will eventually be a single program can be split into several subprograms, each of which can be developed separately. For example, in the case of an automated checkout system the final program may be composed of a supervisory system plus a package of test programs. The detailed requirements for the supervisory system may be relatively insensitive to the object under test and hence can be developed prior to the availability of the complete set of test program requirements. Significant design and development of the final program can then be accomplished prior to the total definition of computer program requirements.

Another approach is to develop enough of the requirements to start design of the program at a high level, detailed design taking place as more requirements become available. For example, design of a program could commence subsequent to the establishment of a set of qualitative performance and design requirements. As more quantitative requirements—accuracies, response times, exact interface data formats, etc.—become known, the specification can be amended allowing the contractor to design the program in more detail. As requirements evolve, the program can be designed, evaluated (such evaluation perhaps aiding in the further definition of requirements) and flow charted. Coding, debugging, and assembly of the program would commence upon release of the completed requirements.

Even where the ultimate form of the system is not known at the inception of design—as in the case of some research and development projects—the requirements can be built up in an orderly fashion. To do this we borrow the concept of a prototype from hardware development. For many systems enough requirements can be developed in the definition phase to serve as the basis for producing a meaningful prototype computer program. This program can then be used in the operational system. As experience is gained by using the prototype, the requirements can be revised, reissued, and from them a "second gen-

*Another benefit of this approach is the isolation of those portions of the program which are expected to change; they in turn can be designed to be flexible, this being stated as a design requirement.
COMPUTER PROGRAM SPECIFICATION  
(Part I)

1. Scope

2. Applicable Documents

3. Requirements  
3.1 Performance  
3.1.1 System Requirements  
3.1.2 Operational Requirements  
1) Function 1  
   1.1 Inputs  
   1.2 Outputs  
   1.3 Information Processing  
2) Function 2  
   2.1 Inputs  
   2.2 Outputs  
   2.3 Information Processing  
n) Function n

3.1.3 Data Base Requirements  
3.1.4 Human Performance

3.2 Computer Program Definition  
3.2.1 Interface Requirements  
3.2.2 User Furnished Programs  
3.3 Design Requirements

4. Quality Assurance  
4.1 Implementation Test Requirements  
4.2 Integration Test Requirements

NOTES

APPENDICES

---

Variations and combinations of the above described approaches may be appropriate in particular efforts. Whatever approach is used, however, a common factor exists: as soon as sufficient requirements have been developed to allow initial design of the program, they are documented and officially released as the "design to" specification. This usually occurs when the performance and design requirements have been defined to a level of detail which permits the contractor to segment the program into component subprograms, allocate functions to the components, and develop flow charts to show the interrelationship among the components.

Control of the specification  
Control of changes and additions to the part I specification is essential if it is to be used as the standard for controlling the design and development of the computer program.  

Formal control, by the user, begins with the release of (some subset of) the requirements as the part I specification. Control is vested in a change control board (CCB) made up of user personnel who are aware of the technical effects of changes throughout the system. Any change to the specification, except for correction of editorial errors, must be approved
The Technical Specification – Key To Management Control

Figure 3 – Definition phase activities leading to the development of the Part 1 specification

by the CCB before the change can be incorporated into the specification. All approved changes must be documented and disseminated to all agencies affected by the changes.†

The CCB must be able to recognize and react to different kinds of changes to the part 1 specification. Many of the early changes to the specification will actually be additions, filling in areas not previously defined. Changes of this type will come from the group in the using agency that developed the original requirements. Changes will also come from other groups in the using agency as a result of further definition of or changes to the system’s requirements. The CCB transmits and coordinates these types of changes to the contractor. Other proposed changes may come from the contractor as he uncovers inconsistencies in the requirements, or formulates new concepts as to what the program should do. If acceptable to the CCB, they are incorporated into the specification.

The specification as a management control tool

Once released, the part 1 specification serves as a valuable tool in controlling the contractor’s design; in assuring the quality of the program, and in accepting the resultant computer program. In many cases it is also a valuable tool in procuring the services of a programming contractor.

Procurement

In those cases where the services of an outside programming contractor are required, the part 1 specification, included as part of a statement of work in a request for proposal, helps define the task to be done. The more complete the specification the better

†A discussion of change control and accounting documents which aid in the change control process is given in the paper by Searle and Neil.
the chances are of getting realistic bids from contractors.

Design control

The specification provides the standard by which the contractor's design may be evaluated. This can be done in several reviews during the programming process, permitting detection of design errors (or faults in the requirements analysis) early enough to do something about them. At least two types of reviews are helpful: a preliminary design review (PDR), and a set of critical design reviews (CDR's).⁴

The PDR is held early in the design process. The contractor's initial overall design approach is reviewed with respect to the part 1 specification. The review gives the user an opportunity to evaluate the proposed design to see if it will meet the requirements as stated in the spec. After the PDR, the contractor proceeds to the detailed design of the individual components of the program.

Once the detailed design of an individual component is complete, it is reviewed in a CDR, prior to coding. This provides an additional check on the ability of the computer program to meet the design and performance requirements. Since the detailed design of a large scale computer program may take considerable time, several CDR's may be held.

Quality assurance

The "quality" of a computer program is determined by its ability to do the job for which it is intended, to be modified as required, to be maintained, and to be understood by others than those who produced it. Although quality in this sense is difficult to measure, it can be enhanced by the availability of programming standards, the application of good documentation techniques, and comprehensive testing of the evolving program. The total specification—both part 1 and part 2—can be used as an instrument in achieving these conditions.

Programming standards are used to control and guide the efforts of programmers during the acquisition phase to ensure that a self consistent computer program is developed (see Figure 4). They are usually of three types: standards of formats, for example, for punched cards and flow charts; standards of design techniques; and standards of procedures regulating programming personnel.⁵ Although the responsibility for developing the standards usually lies with the programming contractor, the requirement for producing them is levied by the user in the design requirements section of the part 1 specification. The user may also include detailed standards in part 1 if appropriate. For example, if the user knows the program will have to be modified extensively in operational use, he could include a requirement in the part 1 specification limiting program module size to 1,000 storage locations.

The part 1 specification also provides a vehicle for indicating the types of tests the contractor must perform during the programming process. These tests are used to determine if the program components are error free, a condition which must be approached if a high quality program is to be produced.

Another essential aspect of quality assurance is documentation of the resultant program. Without adequate documentation, the program cannot be modified or corrected, causing it to lose its value as time progresses. Part 2 of the specification provides a description of the program and as such is the instrument for making and controlling changes to the program in the operation phase. The part 2 specification is a collection of information that is generated during the acquisition phase of any well managed programming effort (see Figure 4). It contains:

1. a description of the overall program design including high level flow charts;
2. programming specifications for the individual components of the computer program;
3. the detailed flow charts for the components;
4. and program listings.

A complete description of a suggested format and content of the part 2 spec is given in the aforementioned government manuals.⁴ ⁵

Acceptance of the program

The role of the part 1 specification in the formal acceptance of the program is discussed elsewhere in this session. Suffice it to say here that part 1 has a twofold role in the acceptance process. It provides the quantitative and qualitative measures by which the performance of the program can be evaluated; it also provides specific requirements for the types of tests and inspections necessary to make this evaluation.

CONCLUDING REMARKS

Computer program requirements, documented in the part 1 specification, provide a standard by which the performance of the contractor and the performance of the resultant computer program can be evaluated.

Despite some inherent problems which may prevent a user from developing complete requirements prior to the design of a computer program, it is possible to use the evolving requirements as a key management tool during the program's development.

An initial part 1 specification can be released early in the programming process for use by a contractor as the basis for the design of the program. The remaining requirements can be transmitted to the contractor in an orderly fashion as they are developed. All sub-
sequent changes to part 1 can be controlled by a change control board, composed of user personnel, to insure the availability of up-to-date requirements consistent with the requirements of any interfacing elements.

Several benefits accrue from having an up-to-date part 1 specification throughout the development cycle of the program. The user can evaluate the contractor's design at periodic intervals. The contractor is provided a basis by which he can develop programming standards, which along with the program description documented in part 2 of the specification is an essential tool in assuring the quality of the program. Also, the user is provided with a measure for determining the acceptability of the program since the result of formal acceptance tests can be compared against the performance requirements in the part 1 specification.

APPENDIX - A model for the Part 1 specification

This appendix presents a more detailed description of the types of information that should be included in a computer program requirements specification. The format illustrated in Figure 2 will be used as the basis for discussion.

Performance requirements (3.1)

The performance requirements put bounds on the expected performance of the program. This is done at several levels. The first level specifies the system characteristics which define the environment that the program operates in (3.1.1). For example, in a space tracking system this could be the number and types of radars involved, the maximum number of space vehicles to be tracked, the numbers and types of displays to be driven, etc. For a payroll program this could include such things as the number of employees, the types of deductions, the salary ranges, etc.
The second level provides an identification of the distinct functions that the computer program must perform. The functions are described, with the aid of a functional block diagram, under the heading "operational requirements" (3.1.2). The block diagram illustrates the relationships of the functions to each other to clarify the textual descriptions that follow in the spec.

The third level provides the inputs, outputs, and required processing for each function depicted in the block diagram. In the space tracking example, as a case in point, the first identified function might be that of orbit determination. A lead paragraph (3.1.-2.1) describes the function in general terms with particular emphasis on its relationship to other functions.

Then the source and type of inputs associated with the function are described (3.1.2.1.1). In our example this might be spacecraft position data from each tracking station and a table of past spacecraft positions generated by the computer program. For each of the input sources, such things as data rates, accuracies, units, and limits should be given.

The types and destinations of data outputs associated with the function are then given (3.1.2.1.2). For example, one such output might be an orbital prediction used for driving a flight controller's display. Again such things as data rates, accuracies, units and limits are specified.

The information processing required for the function is then described (3.1.2.1.3). This is done by prose and mathematical descriptions including necessary logical concepts, timing requirements, formulas and processing accuracies. In our example this might include among other things, the differential equation describing the orbit, the accuracy with which the equation must be solved, and the interval of time within which it must be solved.

The next major class of requirements considered in the performance section of the spec is the requirements for the data base—the collection of data that the program operates on (3.1.3). For many programs the storage requirements for data may exceed the requirements for instructions and must be carefully considered before the program is written. This consideration should include such things as the volume of data, a definition of data classes, for each datum, a description of the units of measure and ranges. If special methods are required to convert the data to a form suitable for use by the computer program, they should be specified. If the program is to be used at several sites where the data values will vary from site to site, the site adaption procedures should be specified.

Many computer programs have extensive human interfaces and like hardware items should be designed with human performance in mind. Therefore, the part 1 specification includes such things as clarity requirements for displays, times for human decision making, maximum display densities, etc. (3.1.4).

**Computer program definition (3.2)**

While the performance requirements may take the bulk of analysis time and occupy most of the content of the specification, there are other important factors that must be considered prior to the design of a computer program. Two of these factors—the interfaces of the computer program with the other elements of the system, and the available computer programs that can or must be used as components of the desired program—are specified under the category, "computer program definition."

The interfaces with other computer programs, communication devices, and equipments must be specified prior to detailed design of the computer program (3.2.1). This includes all relevant characteristics of the computer or class of computers the program is to operate in, including such things as available memory, programming languages, word size, etc. The relationship of the computer program to other equipments and programs should be portrayed in a block diagram, with detailed paragraphs in the spec describing the individual interfaces. The descriptions should include such things as data formats, data rates, and data content. In addition, this portion of the spec should describe all requirements imposed upon other system elements as a result of the design of the program. For example, certain program requirements may affect the design of a display console; if so, they should be documented in the spec.

In many programming efforts there may be available already completed programs which are of potential use to the program designers. For example, a tightly coded numerical routine for solving differential equations may exist in a program library. If there are such programs, the requirements to use them should be stated in the spec (3.2.2).

**Design requirements (3.3)**

There are certain requirements which affect the design of a computer program, which are distinguishable from the performance requirements. These requirements result from general considerations of usability and maintainability, and may include such things as requirements for: the use of programming standards; program organization; special features to facilitate the program's testing; and expandability. These types of requirements should be considered in the definition phase and included in the spec.
Quality assurance (4.0)

One of the major functions of the specification is to define the tests and demonstrations necessary to qualify the program for operational use. For most computer programs the time and resources expended for testing exceed those for any other activity in the programming process. Unless the requirements for testing are considered early enough, there may be delays later in the process when these needs arise. Also, in a large system the requirements for the testing of a computer program are directly related to the requirements for testing the entire system. For these reasons test requirements should be considered in the definition phase and included in the specification.

A method of verification should be given for each of the performance and design requirements stated in the specification. The requirements should be stated to a level of detail which will permit detailed test planning by the program developer. The specification itself should not be a test plan. The types of tests that should be considered include implementation tests, integration tests, preliminary qualification tests, final qualification tests, string tests, etc.³

REFERENCES

1. M V RATYNSKI
   The Air Force Computer Program Acquisition Concept
   SJCC Proceedings 1967

2. L V SEARLE and G NEIL
   Configuration Management of Computer Programs by the
   Air Force Principles and Documentation
   SJCC Proceedings 1967

3. M S PILIGIAN and J L POKORNEY
   Air Force Concepts for the Technical Control and Design
   Verification of Computer Programs
   SJCC Proceedings 1967

4. NPC 500-1
   Apollo Configuration Management Manual (Exhibit XVIII)
   National Aeronautics and Space Administration May 18
   1964

5. ESD EXHIBIT EST-1
   Configuration Management Exhibit for Computer Programs
   Electronic Systems Division Air Force Systems Command
   Laurence G Hanscom Field Bedford Massachusetts
   May 1966

6. B H LIEBOWITZ E B PARKER III and C S SHERRERD
   Procedures for Management Control of Computer Programming in Apollo Appendix I
   TR-66-320-2 Bellcomm Inc September 28 1966

7. IBID APPENDIX VI