Reprinted from *Proceedings, Fifth International Conference on Software Engineering,*

Evaluation of a Software Requirements Document By Analysis of Change Data

Victor R. Basili* and David M. Weiss

Naval Research Laboratory, Washington, D.C. 20375
* Also of the University of Maryland, College Park, Md. 20742

Abstract

We describe in this paper an effective data collection method for evaluating software development methodologies, from definition of the objectives of the data collection to analysis of the results. We show how the data analysis can answer questions with respect to how successfully the goals of the development methodology are met. The A-7 requirements document is used as an example. We provide the results of data analyses conducted partway through the A-7 flight software development cycle, and discuss the utility of information obtained by such partial analyses. Results from the study show that data collection is feasible and useful when performed as part of configuration control, that data distributions based on partial data provide useful feedback to the developers, and that the A-7 Requirements Document is easily maintained and changed.

I. Introduction

In recent years a number of techniques for improving the reliability and decreasing the cost of software have been suggested. These techniques deal with various aspects of the software life cycle. A major issue in software development is the integration of a set of techniques covering one or more phases of the life cycle into a methodology. A second issue is how to refine and adjust the basic techniques of a methodology for the individual factors of some specific environment and application. Software engineering involves the application of a methodology to a particular environment.

The software community is interested in the analysis of techniques, their integration into a methodology, and the engineering of that methodology to particular environments. An effective way to evaluate a methodology, understand the environment, and refine the methodology for the environment is to collect data that characterizes the methodology and the environment and supplies insight into both.

A major source of insight when analyzing a software development project is a record of the changes, including error corrections, made as the development progresses. Data showing where changes were made, what kinds of changes were made, and the effort involved in making changes can be used to evaluate methodologies, characterize environments, and permit the proper engineering of the methodologies for the environments.

This paper has two purposes: (1) to propose an approach for methodology evaluation; the approach is based on goal-directed data collection concurrent with software development, and (2) to show how to apply the approach by example.

The approach described here is being used to evaluate the entire A-7 software redevelopment methodology. The example application is an evaluation of the first stage of the A-7 flight software redevelopment: the production of the A-7 requirements document[7].

We describe the data collection method, from definition of objectives of the data collection to analysis of results. We show how analysis of the data can answer questions with respect to how successfully the goals of the development methodology are met. We provide the results of data analyses conducted partway through the A-7 flight software development cycle, and discuss the usefulness of information obtained by such partial analyses. Section II describes the A-7 flight software development project, its overall objectives, and its requirements document. Section III discusses the relation between the data collection methodology and the software development methodology. Section IV presents the data and its analysis. Section V presents the conclusions and some suggestions for further studies.

II. A-7 Project Overview

A significant obstacle in the field of software engineering is technology transfer. Many techniques are developed in academic environments or in the construction of small programs. Large scale software developers are reluctant to use techniques that have not been tested in the development of complex systems. The Naval Research Laboratory (NRL) and the Naval Weapons Center (NWC) are redeveloping the on-board operational flight program (OFP) for the A-7 aircraft using techniques such as information hiding[11,12], abstract interfaces[14], formal specifications[5,10,16,13], cooperating sequential processes[4,17], process synchronization routines[4], and resource monitors[3,8,9]. (The NWC maintains the current OFP.) The goals are to demonstrate the feasibility of using these techniques to develop a complex, real-time program, and to provide the Navy with a model for the development of avionics programs. The techniques to be used were selected because they are claimed to facilitate the development of software that is reliable, easy to change, easy to understand, and easy to demonstrate correct.

CH1627-9/81/0000/0314$00.75 © 1981 IEEE

The characteristics of the A-7 OFP and the constraints on the redevelopment project are described in Heninger[6]. The program is currently in the design stage. Neither of the authors of this paper were involved in producing the requirements document; one of them has subsequently joined the software design team.

The A-7 Requirements Document

The redevelopment project was started in January 1978. The first task was describing the requirements, which resulted in baselining and publishing the requirements document[7] in November 1978. Approximately 17 man-months of effort were spent in producing the document.

This paper reports on the results of the first major evaluation of the changes to the requirements document. The analyses reported here cover the period from December 1978, when the first change was made to the document, to February 1980. During that time approximately 11 man-weeks were spent in making the 88 changes analyzed for this paper. The total effort spent on A-7 software development activites during that time was 122 man-weeks.

Prior to redesigning the OFP, it was necessary to have a complete, concise description of the requirements for the program. (Although there was an existing program, there was no complete statement of the program's requirements independent of the program.) Four principles were used in designing the requirements document as described by Heninger[6] and Parnas[15]: (1) state questions before trying to answer them, (2) separate concerns, (3) be as formal as possible, and (4) organize according to output of the program.

The resultant document organization is shown in Table 1.

The maintainers of the current OFP at NWC served as consultants to the authors of the document while it was being written. When the authors felt they had a version that was complete and correct, it was subjected to a formal validation review by NWC. After this initial validation, the document was baselined, published, and put under configuration control.

III. Data Collection

Goals

The opportunity to apply recent software engineering technology in the development of a complex model system does not seem to occur often. We considered it important not to lose the chance to monitor closely the progress of the project. A separate data collection effort to permit evaluation of the project during the development cycle was established.

Final evaluation of the success of the rebuilt A-7 software must await the delivery and use of the software. A number of intermediate evaluation points may be established to provide some insight into the redevelopment process. The intermediate evaluations may be based on the goals established at each phase of the project and on the goals established for the different techniques used. As an example, Heninger[6] has described the following six objectives of the requirements document[7].

1. Specify external behavior only.
2. Specify constraints on the implementation.
3. Be easy to change.
4. Serve as a reference tool.
5. Record forethought about the life cycle of the system.
6. Characterize acceptable responses to undesired events.

The main purpose of the data collection and analysis described here is to help measure the success with which the preceding objectives are met. A second purpose is to measure the success with which certain objectives that apply to most requirements document are met. These objectives are listed below.

7. Be correct.
8. Promote error detectability and correctability.
9. Be useful.

Finally, we were interested in learning more about the software development process as expressed by the following additional objectives.

10. Discover effective ways of finding errors.
11. Characterize changes.
12. Characterize errors.

Identification of Data To Be Collected

Once the decision to monitor the project was made and the objectives for the document were clearly stated, the next step was to identify the data to be collected. To do this we established a list of questions[19], the answer to each question helping measure the success of attainment of one or more of the objectives listed in the preceding section. As an example, consideration of the objective "Be easy to change" led to the following questions:

* Is the document easy to change?
* Is it clear where a change has to be made?
* Are changes confined to a single section?

To answer these questions we needed to know, for each change, the effort required to make the change, some measure of how much of the document had to be examined to make the change, and how many sections of the document were actually modified when the change was made. The complete list of questions is shown in Table 2.

Forms Design

Experience gained in designing and using change report forms for NASA's Software Engineering Laboratory[1] and for the Architecture Research Facility study[18] helped considerably in the design of the A-7 change report forms. A prototype form was designed to collect the data needed to answer the questions described in the preceding section and circulated to all members of the A-7 project[19]. The form was modified and the process repeated until all were satisfied with the proposed form. It was then briefly tested, and a few minor modifications made.

Data Collection Procedures

Change data collection was made part of the configuration management process for A-7 documents. As documents are completed, they are placed under configuration control, and all changes made

to them are described and monitored. Change report forms tailored to the objectives and format of the documents under control are used. Figure 1 is the change report form (CRF) used for the requirements document.

Integrating change data collection with configuration control has the advantage that no change data is lost as long as the configuration control process works. Furthermore, one need not have separate forms for configuration control and data collection. The change data analyst is then assured of complete data coverage. In addition, the proposer and implementor of a change are both identifiable if further information is needed.

A characteristic of the A-7 change process is that trails to and from the document and the CRF are maintained; the change data analyst can easily find the exact part(s) of the document changed.

Data Validation and Analysis

Several times a year the accumulated change report forms are reviewed and an analysis conducted for evaluation purposes. As part of this process, the forms are validated. Experience with previous change data collection projects has convinced us that validation of the forms is essential. Validation includes examining each form for completeness and consistency. When necessary, interviews with the proposer and implementor of the change are conducted to obtain missing data and to correct errors.

The various kinds of cross-referencing used facilitate both change to the documentation and change data validation and analysis. As an example, during change validation several incompletely implemented changes have been discovered and reported back to the configuration control board.

IV. Results of the Data Analysis

The answers to the questions posed in the preceding section are presented here, based on data collected during the first fourteen months of use of the requirements document.

Changes discussed in this report fall into one of two categories: error corrections and non-error-corrections. For the sake of brevity, the term error is used in place of error correction, and the term modification is used in place of non-error-correction. The term changes is used to refer to both error corrections and non-error-corrections where no distinction between the two need be made.

The data distributions presented are generally displayed in accordance with the categories used on the CRF. As an example, error distributions use the categories from part 7 of the CRF: "Clerical", "Ambiguity", "Omission", "Inconsistency", "Incorrect Fact", "Information Put in wrong section", "Implementation fact included", and "Other". In cases where there is no data for a particular category, the category is omitted from the figure. For example, Figure 3 does not contain the category "Implementation fact included" because no errors in that category were reported.

Table 3 shows the relationship among (1) the objectives of the requirements document methodology and data collection study, (2) the questions used in measuring attainment of objectives (Table 1), and (3) the figures representing the results of the data analysis. The numbers across the top of the table correspond to the objectives discussed in section III (as an example, objective 1 is "specify external behavior only"). The numbers down the side are the same as those in Table 1. The entries in the table are the numbers of the figures that supply the answers. As an example, Figures 2 and 4 are used to supply the answers to questions 1, 2, 3, and 13, in support of the analysis concerning objective 1. For some questions, there is not yet sufficient data to provide a meaningful answer. The figures that are representative of the analysis that will be done when the data are available are shown for these questions.

Most of the objectives require consideration of several questions, many of the questions are related to attainment of several objectives, and some of the questions require several different analyses (as represented by the figures) to answer. In some cases, answers to questions will become available with further analysis of the data. In others, there is little data as yet on which to base an analysis. Both cases are indicated by asterisks in the table.

At the beginning of a study of changes, it is not always clear what attributes of the changes will be significant. As a result, some of the objectives are stated more precisely (e.g., objectives 3 and 4) than others (e.g., 11 and 12), and the relationship between questions and objectives is not always clear. With this in mind, we have taken a broad view of the relationship between objectives such as 11 and 12 and many of the questions. It is quite possible that the reader's view of these relationships will differ from ours.

Before proceeding to an analysis of each of the questions previously listed, we present some of the general characteristics of the data collected. Figure 2 is a distribution of changes by type. Of the 88 changes reported, 79 were errors. Of the 79 errors, 18 were clerical. Figure 3 shows errors by type.

Figures 2 and 3 show the effort involved in making changes and in correcting errors. Sections of the histograms marked T indicate changes that took a man-hour or less of effort to make (denoted Trivial), those marked E took more than a man-hour but no more than a man-day (denoted Easy), those marked M took more than a man-day but no more than a man-week (denoted Moderate), and those marked F took more than a man-month (denoted Formidable). There were no changes that took more than a man-week but no more than a man-month (denoted Difficult).

As yet, only one Formidable error has been found. This error took six man-weeks of effort to correct, far more than any other change. As a result, it tends to skew effort distributions (e.g., Figure 5). We do not know if this error is an anomaly in terms of effort or not. Where significant, we report results both including and excluding it.

Data on the effort required to understand and make changes is provided in parts 5 and 6 of the CRF (Figure 1). These data are the basis for our effort estimates. The data supplied does not include secretarial and editing effort, only that effort required to understand why a change has to

be made and what change has to be made, and to describe the change in form sufficient for an editor or typist to incorporate it into the document. In addition, nearly all changes were one person changes, i.e. one person noticed the need for the change, did the research necessary to understand what change had to be made, and proposed the change. Nearly all estimates of the effort to make changes can then be viewed as the effort required of one person.

Effort estimates given in this paper are obtained by assuming that Trivial changes took one-half hour of effort, Easy changes one-half day, and Moderate changes one-half week. An estimate of 6 man-weeks for the one Formidable error was obtained through discussions with the person who proposed and implemented the change. It is interesting to note that most of this effort (estimated 80%) involved understanding what the correction should be.

The effort expended in producing the requirements document originally was 17 man-months, including both development and review. The effort expended in making changes is about 11 man-weeks. Of that, about 10.5 man-weeks were spent in correcting errors (4.5 man-weeks without the formidable error). We feel these are small in comparison both to the total effort spent in software development (122 man-weeks) during the time the changes were being made and to the original effort. Discussions with those who wrote and changed the document revealed that many of the people who were making changes were not among the original set of authors; the effort to make the changes consequently contains some learning effort also.

We believe that one reason the requirements document is well-maintained is the ease of making individual changes to it. As will be shown in the discussion of question 4, "typical" changes take about 2.4 hours. These hours are often expended over a relatively long period of time since the need for a change may be noted without specifying the change to be made.

The list of questions below is separated into three categories: (1) those questions for which we believe there is sufficient data to discern patterns in the data distributions (questions with preliminary answers), (2) those questions for which there is insufficient data, and (3) those questions for which lack of data may be meaningful. The questions in the following lists are numbered in accordance with Table 1.

Questions With Preliminary Answers

(2) Are the external interfaces specified correctly?

External interfaces are described in section 2 of the requirements. To answer this question one must find all errors involving section 2. It is also of interest to segregate these errors by type, and to estimate the effort involved in correcting them. Figure 4 shows the distribution of non-clerical external-interface errors by type. Clerical errors have been omitted because we assume that the reason for their occurrence is unrelated to the contents of the section of the document in which they occur.

Section 2 of the requirements is of particular interest because it contained the one Formidable error found so far. This error involved the incorrect definition of a coordinate system. Most of the effort in correcting it was consumed by a study of the use of coordinate systems, the transformations between them, and the sensors providing navigational information for the aircraft. The effort required to correct this error was greater than the effort required to make all other changes to the document combined.

We estimate the effort to correct the non-clerical errors in section 2 of the requirements as 315 man-hours or about 8 man-weeks (2 without the formidable error). This was far more effort than any other section of the document and about 75% of the total effort to correct non-clerical errors so far. One reason for this may be that section 2 has probably received more use as a design specification than any other section at this stage of the project; consequently it has received greater attention than any other section. This issue will not be settled until the project has ended.

(4) Is the document easy to change?

Part 5 of the CRF provides an estimate of the effort to make the change. Using the previously described effort categories and estimation algorithm, based on the responses to part 5 of the CRF we can estimate the effort needed to make changes of various types to the document. The total effort required for all changes estimated in this way is 442 man-hours, or about 11 man-weeks (note that without the one Formidable class error, the effort would be 202 man-hours, or about 5 man-weeks). The average effort to make a change was 5.0 man-hours, and the average to correct an error of any type was slightly higher, 5.4 man-hours. Without the Formidable error, these figures are sharply reduced, becoming 2.3 and 2.4 man-hours respectively. The modes of the effort distributions for changes and errors are both .5 man-hours.

Although there is little data on modifications (only 9 have yet been reported), the initial indication is that they require less effort than errors, averaging 1.8 man-hours.

(5) Is it clear where a change has to be made?

Because of the skewness of the effort distribution, i.e. nearly 70% of the changes in the Trivial category as shown in Figure 5, one might consider the "typical" change as requiring 2.4 man-hours (the average effort omitting the Formidable error). Following Belady and Lehman[2] we will say that a section of the document was handled if it was examined or changed in the course of locating, understanding, or making a change. For all but one change, the set of sections handled was the same as the set of sections changed. We can now characterize the "typical" change as taking 2.4 hours and only requiring handling of the sections of the document that were changed.

(7) Are changes confined to a single section?

Figure 6, obtained from the response to part 2 of the CRF, shows the distribution of changes by the number of sections of the document changed. Most changes only required modifying one section of the document. Analysis of the effort for single section changes compared to multi-section changes shows that on the average the latter required about 27% more effort than the former, 4.8 man-hours for single and 6.1 man-hours for multiple (without the formidable error, about 310% more effort, 1.7 and 6.1 man-hours for single and multiple). Not only

does it seem that the document meets the goal of its authors in this respect, but it also seems that this was a worthwhile goal.

(10) Which sections have the most errors?

Figure 7 shows the distribution of non-clerical errors by section. This distribution shows, for each section, the number of error corrections made in the section. Since one error sometimes resulted in corrections in more than one section, the total number of errors shown here (73) is different than the total number of non-clerical errors (61). The shaded parts of the figure show errors that spanned more than one section. As an example, 18% of the corrections involved the dictionary; about 10% involved the dictionary and other sections (shaded part of the dictionary errors), and 8% involved only the dictionary.

Sections 2 and 4 clearly have the majority of reported errors. This is likely because section 2 has received the most use and section 4 second most at this stage in the development cycle. Further analysis of the data (Figure 8) shows that the distributions of error types in these two sections differ.

(12) Which type of table has the most errors?

Tables tailored to the A-7 flight software are used liberally throughout sections 2, 3, 4, and the dictionary. Four principal kinds of tables are used (see Heninger et al.[7] for definitions of the different types of tables). Of the 61 non-clerical errors so far discovered, 24, or 39%, were errors involving tables. More than half of these (54%) were found in one kind of table; this was the only kind of table for which consistency and completeness checks were not possible.

The distribution of non-clerical table errors by type of error is shown in Figure 9. This distribution differs markedly from the corresponding distribution for all non-clerical errors as shown in Figure 10. Omissions dominate the table errors, whereas incorrect facts dominate the distribution of all non-clerical errors. Furthermore, the margin of domination is smaller for the table errors, and the distribution over omissions, inconsistencies, and incorrect facts is more uniform for the table errors. There are several possible explanations. One is that there is insufficient data yet for the complete pattern to appear. A second is that the tables may be just a good way of organizing information so as to make completeness checks easy.

Because of the relatively small number of non-clerical errors involving tables so far found, it is premature to draw firm conclusions concerning the usefulness of tables in general or event tables specifically from the data. Now that patterns concerning table errors in the partial data have been noticed, we will continue to look for them during the remainder of the development cycle.

Patterns of the sort described in the foregoing provide useful feedback to the developers. Unequal error distributions may mean that some sections of the document have not been as carefully examined or as fully used as others, and require further review.

(14) What use of the document reveals the most errors?

Figure 11 shows the distribution of changes according to how the document was being used when it was discovered that a change had to be made. Since the project is currently in the design phase, it is not surprising that most errors have been discovered as a result of using the document as a software design reference. Recall that data collection started after the document was baselined and had already undergone an initial validation. Clearly, a number of errors remained even after the initial validation process was completed. Some of these, but not the majority, were uncovered by later validation reviews. (This applies especially to one kind of table, for which a good validation algorithm was not discovered until after baselining.)

(15) Are sections 3 (Modes) and 4 (Functions) consistent with each other?

Sections 3 and 4 of the requirements document are complementary views of the system. As yet, there have been only two cases where corresponding errors have been found in both sections.

(16) Is the dictionary complete, correct, and consistent with the rest of the document, and will it remain so?

The dictionary serves as a convenient and useful means for abbreviating and cross-referencing the requirements document. Terms need only be defined in one place, and those unfamiliar with the meaning of a particular term can quickly find its definition. The dictionary also serves as an important tool for abstraction. The definition for a term such as slant range may be used without needing to know how or what data are needed to calculate it.

Figure 12 shows the distribution of changes by section. For each section, the number of changes made in the section is shown. Since one change sometimes resulted in textual changes in more than one section, the total number of changes shown here (102) is different than the total number of changes (88). The shaded parts of the figure show changes that spanned more than one section. As an example, 16% of the changes involved the dictionary; about 10% involved the dictionary and other sections (shaded part of the dictionary errors), and 6% involved only the dictionary.

The dictionary appears to have been well-maintained. Changes elsewhere in the document stimulated the appropriate changes in the dictionary. There seem to be few inconsistencies with the rest of the document.

(18) How is the document being used?

We have not attempted to measure directly the ways in which the document is being used. Assuming that a particular use of the document will reveal errors in proportion to the amount of use, we can infer from Figure 11 the ways in which the document is being used. As might be expected partway through the design stage, the biggest use seems to be as a design reference. There is also a small amount of use by those maintaining the current OFP. Also, after the document was baselined, occasionally a new way of checking completeness or consistency occurred to one of the authors. Such occurrences led to a significant fraction of the changes.

(19) Why are changes being made?

From Figure 2, most changes are made to correct an error in the document. Furthermore, it is rare that a change is incomplete or incorrect. There

has been little need to reorganize the document or to remove information from it.

(20) How many errors are found in the document?

See Figure 2 and previous parts of this section.

(21) What kinds of errors are contained in the document?

Figure 3 shows that most errors found so far are incorrect facts. Based on previous experience, such as reported in Weiss[18], we expected that clerical errors would predominate. The dominant non-clerical errors as shown by Figures 3 and 10 are of a type generally detectable by comparing the document with other sources. Relatively few of them are detectable by self-consistency checks (inconsistencies) or by trying to find alternative meanings (ambiguities). Furthermore, this pattern seems to hold within individual sections, as shown by Figure 8. Preliminary indications are that the authors found it easier to be consistent and precise than complete and correct.

Questions For Which Lack Of Data May Be Meaningful

For some questions, success in attaining the corresponding objectives is indicated by the infrequent occurrence of changes in certain categories. An example is question 1. No errors in the category "implementation fact included" (see section 7 of the CRF, Figure 1b), have yet been reported. This may be an indication that the document does not imply a particular implementation. Similar arguments may be given for questions 2 and 13.

Questions Not Currently Answerable

There is insufficient data available to answer the questions 6, 8, 9, 11, and 17. Requirements for the NRL version of the A-7 OFP were frozen at the start of the redevelopment. Consequently, we do not expect to have data available to answer question 6 until the NRL OFP is completed, when changes to the program will be considered. We expect answers to questions 8, 9, 11, and 17 to start to become available during the detailed design stage, as the document is used more.

V. Conclusions

We have two main objectives in monitoring the changes made to the A-7 software requirements document. One objective is to investigate the feasibility of applying goal-directed data collection concurrently with document maintenance. Similar techniques have been successfully applied to code during program development[18]. A second objective is to try to measure the success with which the A-7 requirements authors met their objectives. We believe the latter objective to be particularly important because the A-7 redevelopers are attempting to use a methodology to produce an engineering model. If they succeed, it is important to know the weak and strong points of the model. If they fail, it is important to know what the troublesome areas are both in the application of particular techniques and in the integration of different techniques.

Two kinds of conclusions may be drawn from this study; one kind concerning the data collection method itself and one kind concerning the A-7 software requirements document. Conclusions concerning the data collection method are listed in the following.

1. The data collection method seems to be feasible and useful. By integrating it with the configuration control process we have tried to keep down the overhead associated with it. Data distributions to answer questions of interest both to the A-7 redevelopers and to software engineers are producible.

2. Data distributions based on partial data seem to provide some useful feedback to the redevelopers. As an example, error distributions that show uneven patterns of error detection may indicate that some document sections need further attention.

3. As patterns are discerned in the data, new questions of interest emerge. As an example, comparison of error distributions across different sections of the document shows that the distributions often differ significantly. There is no obvious explanation for these differences, but many hypotheses can be formed to explain them. We expect that answers to some of the newly-formed questions will be available later in the project; others will probably not be answerable with the data currently being collected.

Conclusions concerning the requirements document are generally answers to the questions discussed in section IV. We list some of the more significant ones below.

1. The document seems to be easily maintained. The low effort to correct a "typical" error supports this conclusion.

2. The document is worth maintaining. The uses to which it is being put as taken from part 1 of the CRF show that it is being heavily used during design.

3. Despite a validation process that included both the authors of the document and the maintainers of the existing flight software, a number of errors remained in the document after it was validated and baselined. The uneven distribution of errors by sections suggest that a significant number of errors may remain in the sections that have been lightly used.

4. The document seems to be well-structured in that changes can be made in one section without requiring many changes elsewhere.

5. The document seems to be relatively more consistent and precise than complete and correct.

We can distinguish between two kinds of effort in maintaining a requirements document. One kind is the effort needed to understand what kind of changes need to be made. The other is the effort in updating the document itself. It is important to note that all the requirements errors must be found in order to produce a correct system whether or not the requirements document is updated to reflect the corrections. As a result, the incremental effort in preserving an accurate requirements document is quite small.

We expect to continue data collection through the entire A-7 flight software redevelopment project. Data collection will be tailored to the project phase and the techniques being used. We have presented here a description of the data collection techniques and results from analysis of partial data because we would like to encourage

others to pursue similar projects.

Acknowledgements

The authors especially thank all of those who have and are continuing patiently to fill out A-7 change report forms, especially Kathryn Heninger and Alan Parker.

We thank Kathryn Heninger, Dr. Rudy Krutar, Alan Parker, Dr. David Parnas, and Dr. John Shore for their suggestions contributing to the design of the requirements document change report form.

Conversations with Kathryn Heninger and Alan Parker were particularly helpful in analyzing the changes.

Finally, Kathryn Heninger, Dr. Carl Landwehr, Greg Lloyd, and the referees provided many comments helpful in improving earlier drafts of this paper.

References

1. V. Basili and M. Zelkowitz, "Analyzing Medium Scale Software Developments," Third International Conference on Software Engineering, Atlanta, Ga., May 1978
2. L. Belady and M. Lehman, "A Model of Large Program Development, IBM Systems Journal, vol. 15, no. 3, 1976
3. P. Brinch Hansen, Operating System Principles, Englewood Cliffs, N. J.: Prentice-Hall, 1973
4. E. W. Dijkstra, "Co-operating sequential processes," in Programming Languages, F. Genuys, Ed. New York: Academic, 1968, pp. 43-112
5. J. V. Guttag, "Abstract data types and the development of data structures," Comm. ACM, vol. 20, pp. 396-404, June 1976
6. K. Heninger, "Specifying Software Requirements for Complex Systems: New Techniques and Their Application," IEEE Trans. Software Eng. vol. SE-6, pp. 2-13, Jan. 1980
7. K. Heninger, J. Kallander, D. L. Parnas, and J. Shore, Software Requirements for the A-7E Aircraft, Naval Res. Lab., Washington, D. C., Memo Report 3876, Nov. 27, 1978
8. C. A. R. Hoare, "Monitors: An operating system structuring concept," Comm. ACM, vol. 17, pp. 549-557, Oct. 1974
9. J. Howard, "Proving Monitors," Comm. ACM, vol. 19, pp. 273-279, May 1976
10. B. Liskov and S. Zilles, "Specification techniques for data abstractions," IEEE Trans. Software Eng., vol. SE-1, pp. 7-19, Mar. 1975
11. D. L. Parnas, "Information distribution aspects of design methodology," in Proc. Int. Fed. Inform. Processing Congr., Aug. 1971, vol. TA-3
12. _____, "On the criteria to be used in decomposing systems into modules," Comm. ACM, vol. 15, pp. 1053-1058
13. _____, "The use of precise specifications in the development of software," in Proc. Int. Fed. Inform. Processing Congr., 1977
14. _____, Use of Abstract Interfaces in the Development of Software for Embedded Computer Systems, Naval Res. Lab., Washington, D. C., Rep. 8047, 1977
15. _____, personal communication, December, 1980
16. D. L. Parnas and G. Handzel, More on Specification Techniques for Software Modules, Fachbereich Informatik, Technische Hochschule Darmstadt, W. Germany, 1975
17. D. L. Parnas and K. Heninger, "Implementing processes in HAS," in Software Engineering Principles, Naval Res. Lab., Washington, D. C., course notes, 1978, Document HAS.9
18. D. M. Weiss, "Evaluating software development by error analysis: the data from the Architecture Research Facility," Journal of Systems and Software, vol. 1, pp. 57-70 1979
19. _____, Design of the A-7 Requirements Document Change Report Form, NRL Technical Memorandum 7503-320, December, 1978

0. Introduction: A description of the document organization, an abstract for each section, and a guide to the notation used.
1. Distinguishing Characteristics of the TC-2 Computer
2. Input and Output Data Items: Description of the information received and transmitted by the computer, organized according to device, one subsection per device connected to the computer.
3. Modes of Operation: States of the program corresponding to aircraft operating conditions.
4. Time-independent Description of A-7 Software Functions: Each function description characterizes one or more output data items and specifies the conditions under which they are updated.
5. Timing Requirements: Timing requirements for all functions described in section 4.
6. Accuracy Constraints on Software Functions
7. Undesired Event (UE) Responses: Desired behavior of the system when undesired events occur.
8. Required Subsets: Useful subsets of the system obtainable by omitting parts of the code.
9. Possible changes: Possible future modifications to the OFP.
10. Glossary: Glossary of acronyms and technical terms used by the A-7 community.
11. References

Indices: Alphabetical indices to data item descriptions, mode overviews, and functions.

Dictionary: Definitions of standard terms used in the mode (section 3) and function (section 4) descriptions.

Table 1 Sections of the Requirements Document

(The preceding section descriptions are either taken from or follow closely the descriptions given in Heninger[6] and Heninger et al.[7].)

1. Is externally-visible behavior only specified without implying a particular implementation?
2. Are the appropriate external interfaces specified?
3. Are the external interfaces specified correctly?
4. Is the document easy to change?
5. Is it clear where a change has to be made?
6. Are the changes likely to occur predicted correctly?
7. Are changes confined to a single section?
8. Is the proper set of undesired events described?
9. Is the notation used unambiguous?
10. Which sections have the most errors?
11. Where do the most changes have to be made?
12. Which type of tables has the most errors?
13. Does the document contain unnecessary information?
14. What use of the document reveals the most errors?
15. Are sections 3 (Modes) and 4 (Functions) consistent with each other?
16. Is the dictionary complete, correct, and consistent with the rest of the document, and will it remain so?
17. Which subsections of sections 2 (Data Items), 3 (Modes), and 4 (Functions) are most error-prone?
18. How is the document being used?
19. Why are changes being made?
20. How many errors are found in the document?
21. What kinds of errors are contained in the document?

Table 2. Questions Used In Designing The Change Report Form

A-7 Requirements Document Change Report Form

Number _____

Current Date..... Month | Day | Year
Change Started On..

Change Discovery

1. How was the document being used when the need for the change was discovered?
   - ☐ Validation review by the authors
   - ☐ Validation review by non-authors
   - ☐ As a maintenance reference
   - ☐ As a software design reference
   - ☐ As a coding reference
   - ☐ As a test reference
   - ☐ Other: _____

Identification

2. Description of change: _____

3.

Section(s) Changed		Section(s) Examined But Not Changed
Name	Section	Subsection(s)	Subsection(s)
Intro.	0.		
TC 2	1.		
Data Items	2.		
Modes	3.		
Functions	4.		
Timing	5.		
Accuracy	6.		
UE's	7.		
Subsets	8.		
Changes	9.		
Glossary	10.		
Sources	11.		
Data Item Index			
Mode Index			
Function Index			
Dictionary			

Type Of Change

4. Why is the change being made (check one)?
   - ☐ To correct an original error
   - ☐ To complete or correct a previous change (Previous CRF # _____)
   - ☐ To comply with unexpected requirement change (violates sect. 9 assumption)
   - ☐ To comply with expected requirement change (assumed in sect. 9 subsect. __)
   - ☐ To remove unnecessary information
   - ☐ To reorganize within one section } Readability Improvement
   - ☐ To reorganize among several sections } Readability Improvement
   - ☐ Other: _____

5. What was the effort in person-time required to understand and make the change?
   0 — 1 man-hr — 1 man-day — 1 man-week — 1 man-month

6. Estimate the percentage of the effort just to understand the change.
   0 — 20 — 40 — 60 — 80 — 100

CHANGE REPORT FORM (CRF) OBVERSE
FIGURE 1a

FOR ERROR CORRECTIONS ONLY

Type Of Error

7. How is the error best characterized (check one)?
   - ☐ Clerical
   - ☐ Ambiguity
   - ☐ Omission
   - ☐ Inconsistency
   - ☐ Incorrect Fact
   - ☐ Information put in wrong section } Inappropriate Fact
   - ☐ Implementation fact included } Inappropriate Fact
   - ☐ Other: _____

For Section 2 (Data Item) Errors

8. Which part(s) of the subsection were incorrect (check all that apply)?
   - ☐ Hardware
   - ☐ Description
   - ☐ Value Characteristics
   - ☐ Instruction Sequence
   - ☐ Data Representation
   - ☐ Comment

For Section 3 (Modes) and Section 4 (Functions) Errors

How is the error best characterized (check all that apply)?

9. Section 3
   - ☐ Mode Transition Error
   - ☐ Mode Condition Error
   - ☐ Mode Overview Error

10. Section 4
   - ☐ Applicable Mode List Error
   - ☐ Output Data Item Error
   - ☐ Output Description Error
   - ☐ Event Table Error
   - ☐ Selector Table Error
   - ☐ Condition Table Error
   - ☐ Inconsistent or incomplete table

11. ☐ Corresponding errors were made in both sections 3 and 4.

For Errors Involving Dictionary Items

12. How is the error best characterized (check all that apply)?
   - ☐ Incorrect item
   - ☐ Incomplete item
   - ☐ Dictionary inconsistent with usage elsewhere

For Notational Errors

13. Which notation type was in error?
   - ☐ /input/
   - ☐ //output//
   - ☐ $value$
   - ☐ !item!
   - ☐ *mode*
   - ☐ operator
   - ☐ event
   - ☐ simple condition
   - ☐ compound condition

14. How is the error best characterized?
   - ☐ incorrect grouping
   - ☐ incorrect symbol
   - ☐ incorrect value

Comments

15. Please give any information that may help understand the change and its cause.

_____ Name _____ Date

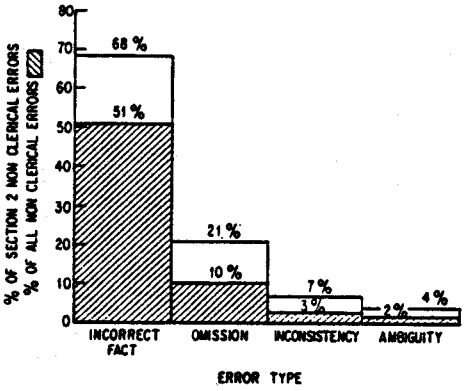
CHANGE REPORT FORM (CRF) REVERSE
FIGURE 1b

Question/	Objective 1	2	3	4	5	6	7	8	9	10	11	12
1	2,3						2,3				2,3	2,3
2	4	4					4				4	4
3	4,7	4,7					4,7				4,7	4,7
4			2,5					2,5			2,5	
5			2,5,6					2,5,6			2,5,6	
6			2		2		2	2			2	
7			6,12					6,12			6,12	
8						12	12				12	12
9				*			*	*			*	*
10							7				7	7
11					12						12	
12							9	9		9	9	9
13	2,3,10						2,3,10				2,3,10	2,3,10
14			*					*	*	*	*	*
15			*				*	*		*	*	*
16				12,*			12,*	12,*			12,*	12,*
17	*						*				*	*
18									11	11		
19					2						2	
20							2,3	2,3		2,3	2,3	
21							3,8-10				3,8-10	3,8-10

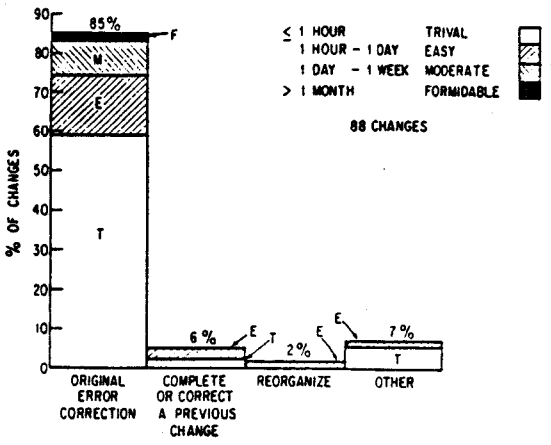
* Analysis or data not yet available, or further analysis may provide more information.

Table 3 Relationship Among Objectives, Questions, and Figures

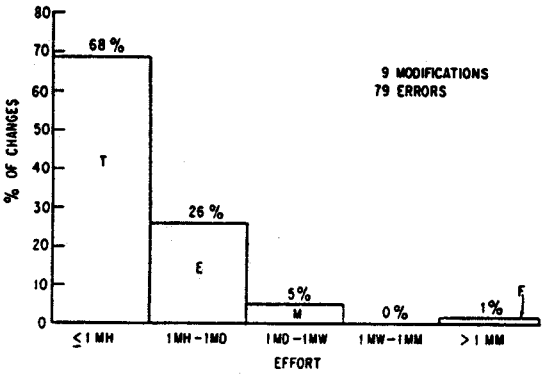
(Entries are figure numbers)



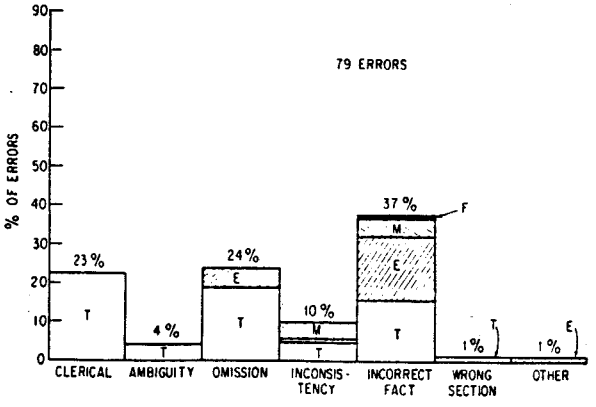
NON-CLERICAL ERRORS IN SECTION 2
FIGURE 4



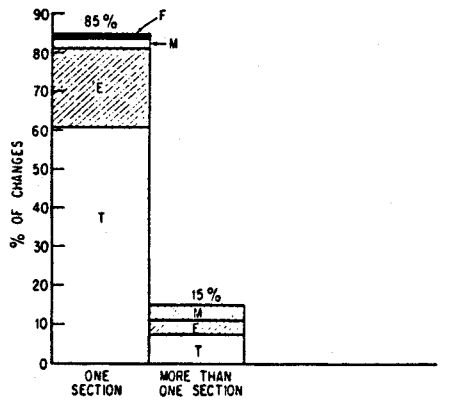
TYPES OF CHANGES
FIGURE 2



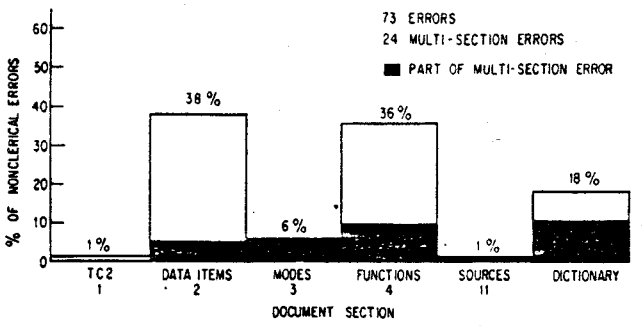
EFFORT TO CHANGE
FIGURE 5



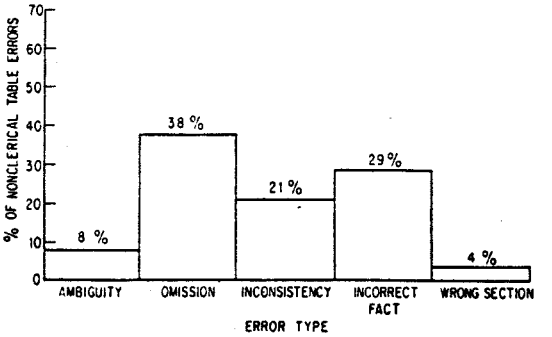
TYPES OF ERRORS
FIGURE 3



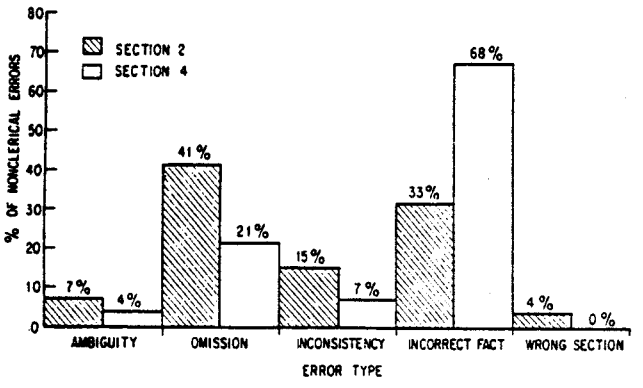
CONFINEMENT OF CHANGES
FIGURE 6



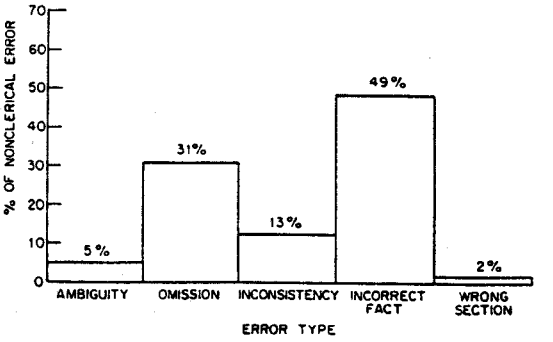
NON-CLERICAL ERRORS BY SECTION
FIGURE 7



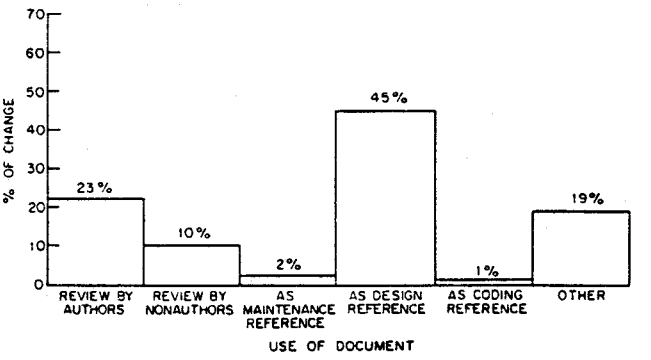
NON-CLERICAL TABLE ERRORS
FIGURE 9



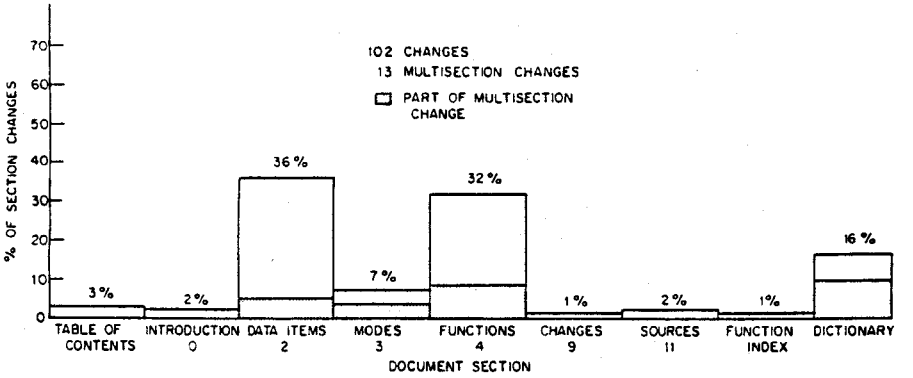
NON-CLERICAL ERRORS IN SECTIONS 2 AND 4
FIGURE 8



NON-CLERICAL ERRORS BY TYPE
FIGURE 10



DISCOVERY OF NEED FOR CHANGE
FIGURE 11



CHANGES BY SECTION
FIGURE 12