SOFTWARE MAINTENANCE WORKSHOP

RECORD

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA
DECEMBER 6-8, 1983

```
1600 D.T=12: M.T=(WD-5)*D.T-1
1610 LOCATE NL-1,3 : PRINT STRING$(WD-5,BULLIT$);
1620 FOR I=3 TO WD-3 : BUL(I)=1 : NEXT : NB=WD-5
1630 FOR I=2 TO NL-1
1640   LOCATE I,1 : PRINT CHR$(186); : LOCATE I,WD-1 : PRINT CHR$(186);
1650 NEXT
1660 LOCATE 1,1 : PRINT CHR$(201);STRING$(WD-3,205);CHR$(187);
1670 LOCATE NL,1 : PRINT CHR$(200);STRING$(WD-3,205);CHR$(188);
1680 LOCATE NL-2,1 : PRINT CHR$(199);STRING$(WD-3,196);CHR$(182);
1690 BX=WD/2 : LOCATE NL-2,BX : PRINT CHR$(208);
1700 LOCATE 1,8*(WD/40):PRINT " SCORE   ";SCORE:LOCATE 1,26*(WD/40):PRINT
1710 FOR C.T=1 TO M.T
1720   FOR I=3 TO M.S
1730     ON S.STAT(I)+1 GOTO 1740,1800,1930,1960,2080,2120
1740     ' Inactive
1750     IF RND).17 THEN 2190
1760     J=4*RND
1770     S.STAT(I)=1 : S.X(I)=3 : S.Y(I)=I : S.SP(I)=N.SP(J) : S.PAT(I)=0
1780     S.PIC$(I,0)=N.PIC$(J,0):S.PIC$(I,1)=N.PIC$(J,1) : S.SCORE(I)=N.S
1790     S.LEN(I)=N.LEN(J):LOCATE S.Y(I),S.X(I)-1:PRINT S.PIC$(I,0); : GO
1800     ' Fly
1810     X1=S.X(I) : Y=S.Y(I) : X2=S.X(I)+S.LEN(I)
1820     IF C.T AND S.SP(I) THEN X2=X2-1 : GOTO 1860
1830     LOCATE S.Y(I),X1
1840     PRINT S.PIC$(I,S.PAT(I)); : X1=X1+1 : S.X(I)=X1
1850     S.PAT(I)=1-S.PAT(I) : IF X1=WD-7 THEN S.STAT(I)=2
1860     FOR J=0 TO M.B
1870       IF B.ACT(J)=0 THEN 1910
1880       IF B.Y(J)()Y THEN 1910
1890       IF B.X(J))=X1 THEN IF B.X(J)(=X2 THEN 1900 ELSE 1910 ELSE 1910
1900       S.STAT(I)=3 : B.ACT(J)=0 : SCORE=SCORE+S.SCORE(I)
1905       REM  WORKSHOP ON SOFTWARE MAINTENANCE 1983
1910     NEXT J
1920     GOTO 2190
1930     ' Final
1940     LOCATE S.Y(I),S.X(I) : PRINT "      "; : S.STAT(I)=0
1950     GOTO 2190
1960     ' Hit
1970     PLAY SND$(S.LEN(I)-1)
1980     X=S.X(I) : Y=S.Y(I) : LN=S.LEN(I)
1990     IF CLR THEN COLOR 4
2000     LOCATE Y-1,X-1 : PRINT X.TOP$(LN);
```

ISBN 0-8186-0510-3
LIBRARY OF CONGRESS NO. 83-82757
IEEE CATALOG NO. 83CH1982-8
IEEE COMPUTER SOCIETY ORDER NO. 510

Edited by: Dr. Robert S. Arnold, Software Consultant

Sponsored by:

IEEE Computer Society Technical Committee on Software Engineering

IEEE Computer Society

National Bureau of Standards

Naval Postgraduate School

In cooperation with:

ACM Special Intererest Group on Software Engineering (SIGSOFT)

E OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.
IEEE COMPUTER SOCIETY
IEEE COMPUTER SOCIETY PRESS

TOWARDS MAINTAINABILITY THROUGH SPECIFICATIONS QUALITY*

ANA REGINA CAVALCANTI DA ROCHA
Instituto Militar de Engenharia, Rio de Janeiro-BRAZIL

ARNDT von STAA
Pontifícia Universidade Católica, Rio de Janeiro-BRAZIL

ABSTRACT

A substantial part of corrective maintenance is due to inadequate specifications. A large portion of the errors are due to low quality specifications. This paper defines a model for evaluating specifications quality. Using this model, the expected amount of corrective maintenance will be reduced. It is also expected that the effort spent in evolutionary maintenance will be reduced.

INTRODUCTION

Comments about difficulties in developing software are commonplace. Budgets and schedules are frequently overrun and many times the product does not meet user needs leading, thus to "corrective maintenance". It is a well known fact that a substantial part of corrective maintenance is due to inadequate specifications[1]. Part of these inadequacies are obviously due to our difficulty in achieving a full understanding of the problem to be solved. However, a large portion of the errors are due to low quality specifications.

Specifications are documents produced during the construction of software systems. These documents (requirements specifications, design specifications, program specifications, data specifications, module specifications) describe the system and, later, its components, starting from a macroscopic view and progressing to more detailed ones. Specifications are intended to perfectly describe the essential features, characteristics and requirements of the system and each of its components. Clearly software quality depends on the quality of its specifications. Thus, it is essential to produce high quality specifications.

Quality, however, is a multidimensional concept. The quality of specifications is resultant of many attributes. Consequently, in order to be able to measure specifications quality we must first determine what characteristics to measure. That is, we have to identify the attributes which determine specifications quality.

THE PROCESS OF PRODUCING SPECIFICATIONS

In order to identily relevant quality attributes for specifications we must understand how specifications are constructed and used. Several activities are involved when producing specifications (fig. 1):

1. evolution, which employs user's knowledge (needs, expectations, etc) and specifier's knowledge (ideas, experience, standards, etc) to either create a fully new specification or to evolve from an already existing document to create a more detailed specification;

2. modification, which applies change proposals to already existing specifications;

3. reflection, which applies changes or even produces a new specification at a more macroscopic level in order to accomodate restrictions and changes made on more detailed specification;

These activities are seldom performed with perfection. So two other activities become necessary - verification and validation - which aim at identifying imperfections. Thus we have:

4. verification, which evaluates the specification quality with regard to its intrinsic attributes;

5. validation, which evaluates the achievability of the system's goals, assuming it is implemented as specified.

In order to be able to manage and to enhance the specification process itself, this process must be evaluated. Thus we have:

6. monitoring, which gathers data about the previously described processes' performance;

7. management, which acts upon the whole development process.

A MODEL FOR EVALUATING SPECIFICATIONS QUALITY

High quality specifications must support the activities described above. As already said, this requires the presence of several attributes. We need, thus, a model to organize these attributes and which allows us to synthesize all measures obtained. Using a terminology similar McCall[2], we define the following concepts:

* Research supported by FINEP contract 3.2.82.0179.00

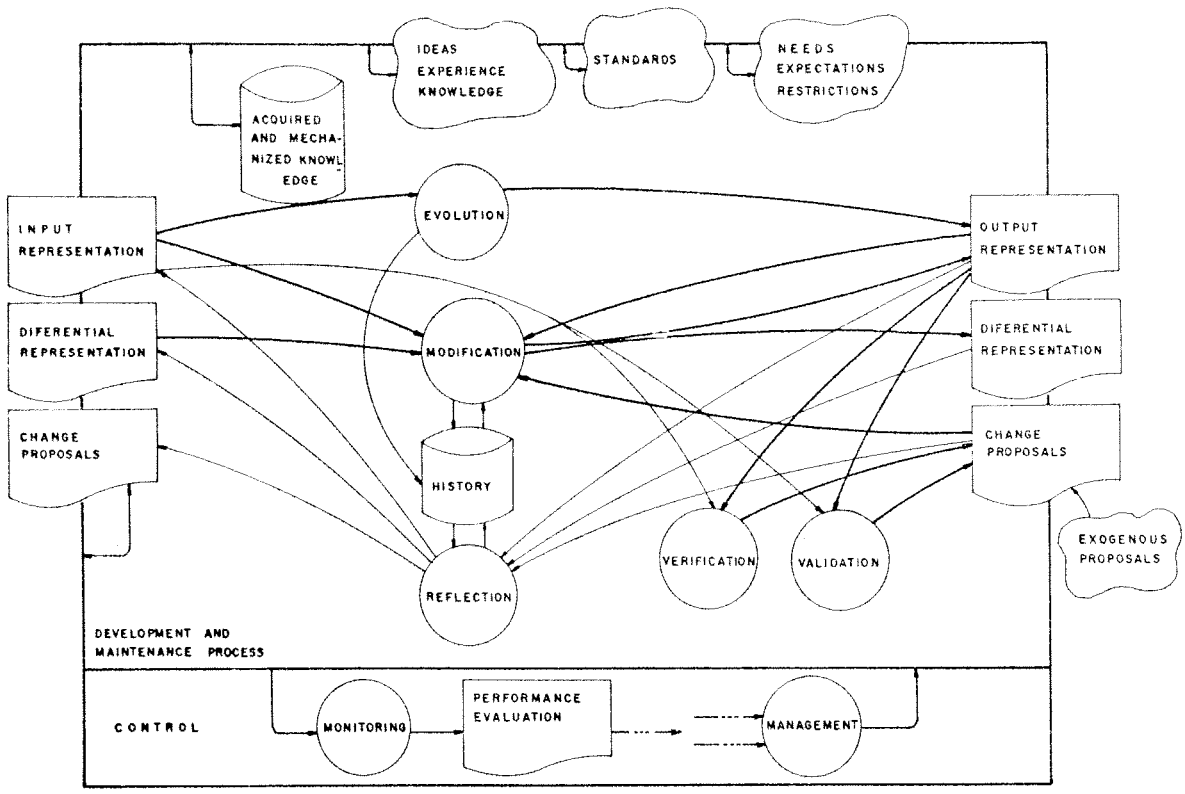

FIG. 1 - ACTIVITIES PERFORMED DURING THE SPECIFICATION PROCESS

1. specification quality objectives: determine the overall expectations which specifications should satisfy;

2. specification quality factors: determine the specifications quality from the point of view of the user (system users and developers) of these specifications;

3. criteria: define specific attributes to be measured or evaluated, determine the process to be used in order to measure or evaluate these attributes and establish the instruments to be used;

4. metrics: are quantitative measures obtained when applying criteria to evaluate a given specification;

5. normalization functions: which valuate factors from a given set of metrics.

Figure 2 shows the relation among these concepts.

The specification quality objectives are directly related to the activities of the specification process. Each objective corresponds to one activity. Thus, specifications should satisfy the following objectives: ability to evolve, modifiability, reflection aids, verifiability and validability.

Objectives are achieved through the specification quality factors. Specification quality factors are defined by means of sub-factors and criteria. Normalization functions determine a value - degree of presence - of each factor. Normalization functions depend on values - metrics - gathered by applying criteria to a given specification. Each criterion defines what is being evaluated, how the evaluation is to be performed and the scale (set of values on units of measure) to be used.

Metrics are, thus, values resulting from the evaluation of the product in accordance to a specific criterion. Frequently this evaluation must be based on opinions, gathered by experts, due to a lack of more objective evaluation methods. In this case, a scale varying from 0 to 10 may be used, where 10 means the best possible situation with regard to the attribute evaluation and 0 the worst. To reduce subjectivity in the evaluation, characterizations of points on the scale should be provided[3].

Normalization functions are used to valuate a given factor, synthesizing a set of metrics and possibly sub-factors. Generally they are weighted means of metrics normalized to the interval 0 to 1, where 0 means that the factor is nonexistent and 1 that it is completely satisfied. With this convention it is possible to define acceptability levels for the factors.

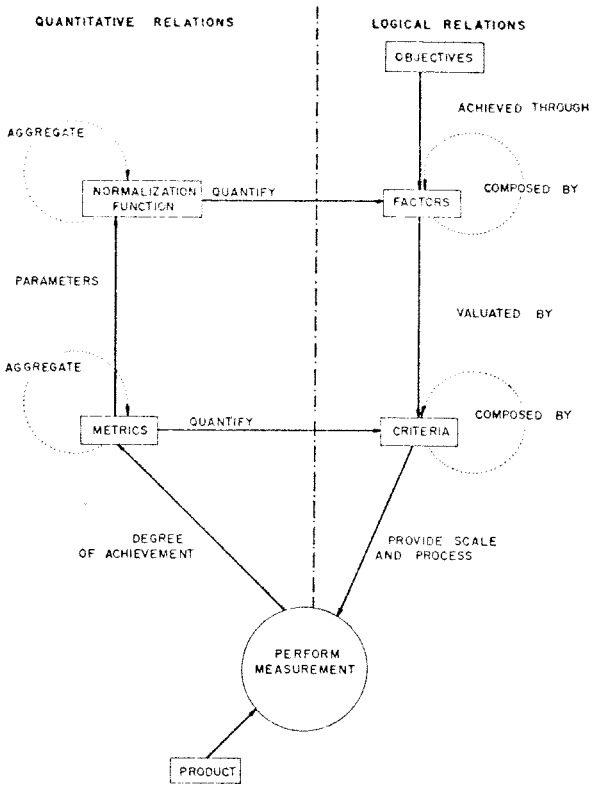

FIG. 2 - THE QUALITY MODEL

Using this quantitative model we are now able to specify the required level of quality which specifications should achieve. Consequently, we may balance costs to achieve quality with the estimated value of this quality (reduced number of erros, reduced costs to recover from failures, reduced effort to maintain, etc). Furthermore, the quality expectancy is increased, since now specifiers know in advance how careful they have to work and how the results of their efforts will be evaluated. Thus the "quality consciousness" is greatly increased.

Figure 3 shows the specification quality factors and their related criteria. It is beyond the scope of this paper to describe in details each of these factors and criteria.

Some comments are due, though, especially with regard to the factor "adequability". This factor obviously depends on the product (system, program, module, data abstraction, etc) being specified. Hence, precisely defined criteria can only be elaborated during the evolution of the project.

Some experimental evidence has been gathered[4,5]. These experiments, have shown that the method is viable and effective. However, more evidence must be gathered (research and experiments are underway), specially with regard to the importance of each of the factors and criteria, as well as with regard to the "evaluateability" of each identified criterion.

CONCLUSION

With the utilization of the quality specification and evaluation model described in this paper, specifications of acceptable levels of quality must be achieved. Consequently we will have a better control of software quality. Thus the expected amount of corrective maintenance will be reduced, and the effort spent on evolutionary maintenance will be reduced too.

The quality model described in this paper is generic. Its scope is wider than usually needed. Thus an adequated subset should be chosen for each project. The model is also flexible enough to evolve with software engineering practice. A full description of the model exceeds the scope of this paper. A detailed description can be found in[5].

REFERENCES

[1] B. Boehm, *Software Engineering Economics*. Prentice-Hall, 1981.

[2] J. McCall, "An introduction to software quality metrics", *Software Quality Management*, J.D. Cooper and M.J. Fisher (eds), Petrocelli, 1979.

[3] D.E. Peercy, "Software maintainability evaluation methodology", IEEE Transactions on Software Engineering, vol. SE-7, n. 4, 1981.

[4] J.C. Barros, "Um modelo para avaliação da quali

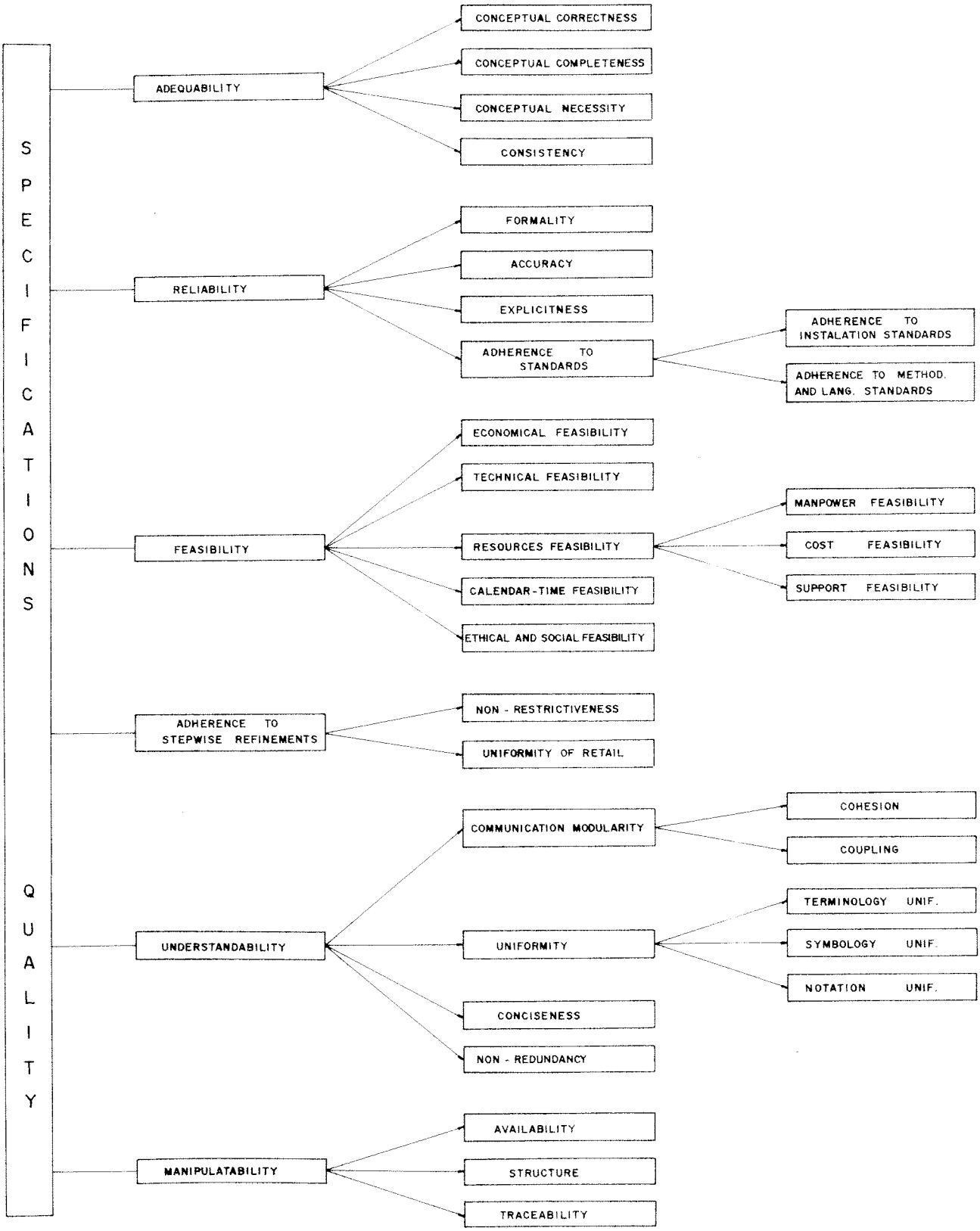

FIG. 3 - SPECIFICATION QUALITY FACTORS

dade de especificações; um experimento", Master's Dissertation, Universidade Federal de Pernambuco, Recife, Brazil, 1983 (in portuguese).

[5] A.R.C. da Rocha, "Um modelo para avaliação da qualidade de especificações"; PhD Thesis, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Brazil; 1983 (in portuguese).